2017 IEEE 14th International Scientific Conference on Informatics

INFORMATICS 2017

November 14-16, 2017, Poprad, Slovakia

PROCEEDINGS

Editors
Valerie Novitzká
Štefan Korečko
Anikó Szakál

Organized by

Affiliated branch of the Slovak Society for Applied Cybernetics and Informatics at Department of Computers and Informatics, FEEI TU of Košice

Faculty of Electrical Engineering and Informatics, Technical University of Košice

Association of Slovak Scientific and Technological Societes

IEEE SMCS Technical Committee on Computational Cybernetics

ISBN 978-1-5386-0888-3
IEEE catalog number CFP17E80-PRT

2017 IEEE 14th International Scientific Conference on Informatics

IEEE Catalog Number CFP17E80-PRT
ISBN 978-1-5386-0888-3

Additional copies of this publication are available from
Curran Associates, Inc.
57 Morehouse Lane
Red Hook, NY 12571 USA
+1 845 758 0400
+1 845 758 2633 (FAX)
email: curran@proceedings.com

Preface

It is with pleasure that we hereby present the proceedings of the *2017 IEEE 14th International Scientific Conference on Informatics*. Its success builds upon our ever improving efforts to publish with higher standards in the various areas of informatics. As part of this, our conference becomes a significant international forum for presenting original research results, sharing experience, and exchanging new ideas. The topics of this conference cover theoretical and practical results, along with methods for transferring these research results into real-life domains, by scientist and experts working in computer science and informatics. The conference also provides an opportunity for young researchers to demonstrate their achievements and to discuss their results at an international scientific forum. The main topics of the conference are

- Computer Architectures,
- Computer Networks,
- Theoretical Informatics,
- Programming Paradigms and Languages,
- Software Engineering,
- Distributed Systems,
- Computer Graphics and Virtual Reality,
- Artificial Intelligence,
- Data and Knowledge Bases,
- Intelligent Information Systems,
- Applied Informatics and Simulation and
- Quality of Software and Services.

The proceedings start with keynote lectures from three eminent experts in their respective fields. Every paper in these proceedings has been peer-reviewed by two independent external referees. On behalf of the Programme and Organizing Committees, we would like to thank all the reviewers for their time and effort in reviewing the papers. Their contribution has ensured the high quality of publications in these proceedings. We would also like to extend our thanks to all the authors and keynote speakers, who contributed and guaranteed the high and professional standard of this conference.

The *2017 IEEE 14th International Scientific Conference on Informatics* has been organized by

- Faculty of Electrical Engineering and Informatics, Technical University of Košice,
- Slovak Society for Applied Cybernetics and Informatics at Department of Computers and Informatics,
- Association of Slovak Scientific and Technological Societies and
- IEEE SMCS Technical Committee on Computational Cybernetics.

We also thank the sponsors

- IEEE Hungary Section,
- IEEE SMC Chapter, Hungary,
- IEEE Joint IES/RAS Chapter, Hungary

and the technical co-sponsor

- IEEE SMC Society.

The conference is held in Poprad, a historical city nestled at the foot of the High Tatra Mountains, a region rich in culture and natural beauty. We are confident that inside these covers, you will find papers relevant to your field of interest. We also look forward to your participation at the next event of this conference.

Poprad, November 2017

On the behalf of the Programme and Organizing Committees
Valerie Novitzká

Committees

General Chair

prof. Ing. Liberios Vokorokos, PhD., *Dean of Faculty of Electrical Engineering and Informatics, Technical University of Košice, SK*

Honorary Chair

akad. prof. Ing. Ivan Plander, DrSc., *Slovak Society for Applied Cybernetics and Informatics, SK*

Program Chair

Valerie Novitzká, *Technical University of Košice, SK*

Program Committee

Miklós Bartha, *Memorial University of Newfoundland, CA*
Andreas Bollin, *University of Klagenfurt, AT*
Dmitriy B. Buy, *National University of Taras Shevchenko, Kyiv, UA*
Jan Čapek, *University of Pardubice, CZ*
Svetlana Cicmil, *University of the West England, Bristol, UK*
Zbigniew Domański, *Częstochowa University of Technology, PL*
Erik Duval, *Katholieke Universiteit Leuven, BE*
Dimitar Filev, *Ohio State University, US*
Zoltán Fülöp, *University of Szeged, HU*
Gianina Gábor, *University of Oradea, RO*
Ján Genči, *Technical University of Košice, SK*
Andrzej Grzybowski, *Częstochowa University of Technology, PL*
Klaus Haenssgen, *Leipzig University of Applied Sciences, DE*
Tamás Haidegger, *Óbuda University, Budapest, HU*
Aboul Ella Hassanien, *Cairo University, Giza, EG*
Zdeněk Havlice, *Technical University of Košice, SK*
Pedro Rangel Henriques, *University of Minho, Braga, PT*
Pavel Herout, *University of West Bohemia, Pilsen, CZ*
Ladislav Hluchý, *Slovak Academy of Sciences, Bratislava, SK*
Elke Hochmüller, *Carinthia University of Applied Sciences, AT*
László Horváth, *Óbuda University, Budapest, HU*
Zoltán Horváth, *Lóránd Eötvös University, Budapest, HU*
Péter Kádár, *Óbuda University, Budapest, HU*
Waldemar W. Koczkodaj, *Laurentian University, CA*
Levente Kovács, *Óbuda University, Budapest, HU*
Jiří Kunovský, *Brno University of Technology, CZ*
Sandra Lovrenčić, *University of Zagreb, HR*
Mária Lucká, *Slovak University of Technology, Bratislava, SK*
Ivan Luković, *University of Novi Sad, RS*
Dragan Mašulović, *University of Novi Sad, RS*
Karol Matiaško, *University of Žilina, SK*
Marjan Mernik, *University of Maribor, SI*
Jurij Mihelič, *University of Ljubljana, SI*
Hanspeter Mössenböck, *Johannes Kepler University Linz, AT*
Günter Müller, *Albert-Ludwigs-Universität Freiburg, DE*
Hiroshi Nakano, *Kumamoto University, JP*
Mykola S. Nikitchenko, *National University of Taras Shevchenko, Kyiv, UA*
Lucia Pomello, *University of Milano-Bicocca, IT*
Herbert Prähoffer, *Johannes Kepler University Linz, AT*
Horia F. Pop, *Babes Bolyai University, Cluj, RO*
Stanislav Racek, *University of West Bohemia, Pilsen, CZ*
Mihály Réger, *Óbuda University, Budapest, HU*
Sonja Ristić, *University of Novi Sad, RS*

Imre J. Rudas, *Óbuda University, Budapest, HU*
Gábor Sági, *Hungarian Academy of Sciences, Budapest, HU*
Abdel-Badeeh M. Salem, *Ain Shams University, Cairo, EG*
Wolfgang Schreiner, *RISC, Johannes Kepler University Linz, AT*
Elena Somova, *University of Plovdiv, BG*
William Steingartner, *Technical University of Košice, SK*
Jiří Šafařík, *University of West Bohemia, Pilsen, CZ*
Petr Šaloun, *VŠB - Technical University of Ostrava, CZ*
Jarmila Škrinárová, *Matej Bel University, Banská Bystrica, SK*
Michal Štepanovský, *Czech Technical University in Prague, CZ*
József K. Tar, *Óbuda University, Budapest, HU*
Katarína Teplická, *Technical University of Košice, SK*
Renáta Tkáčová, *Technical University of Košice, SK*
Tsuyoshi Usagawa, *Kumamoto University, JP*
Valentino Vranić, *Slovak University of Technology, Bratislava, SK*
Neven Vrček, *University of Zagreb, HR*
František Zbořil, *Brno University of Technology, CZ*
Jaroslav Zendulka, *Brno University of Technology, CZ*
Doina Zmaranda, *University of Oradea, RO*

Organizing Committee

Milan Šujanský, *Technical University of Košice, SK* (chair)
Anikó Szakál, *Óbuda University, Budapest, HU* (financial chair)
William Steingartner, *Technical University of Košice, SK* (general manager)
Martina Dragošeková, *Technical University of Košice, SK*
Sergej Chodarev, *Technical University of Košice, SK*
Štefan Korečko, *Technical University of Košice, SK*

Technical staff

Ján Perháč, *Technical University of Košice, SK*
Jana Šťastná, *Technical University of Košice, SK*

ACKNOWLEDGEMENT TO REVIEWERS

We would like to gratefully appreciate the following distinguished reviewers for spending their invaluable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of the 2017 IEEE 14th International Scientific Conference on Informatics:

Norbert Ádám
Marián Klešč
Piotr Puchała
Mikuláš Alexík
Ján Kollár
Kornelije Rabuzin
Gabriela Andrejková
Štefan Korečko
Danijel Radošević
Michaela Bačíková
Paweł Kossecki
Sonja Ristić
Anton Baláž
Mariusz Kubanek
Krzysztof Rojek
Igor Banduric̆
Mirosław Kurkowski
Gábor Sági
Ľubomír Beňa
Temur Kutsia
Wolfgang Schreiner
Günter Blaschek
Dominik Lakatoš
Olga Siedlecka-Lamch
Janusz Bobulski
Kornélia Lazányi
Vladimír Siládi
Ján Buša
Sandra Lovrenčić
Branislav Sobota
David Cerna
Ivan Luković
Michael Sonntag
Zbigniew Domański
Pavol Macko
William Steingartner
Peter Drotár
Branislav Madoš
Łukasz Szustak
Peter Feciľak
Dragan Mašulović
Petr Šaloun
Lidija Fodor
Klaus Miesenberger
Václav Šátek
Zoltán Fülöp
Juraj Mihaľov
Slavomír Šimoňák
Juraj Gazda
Jurij Mihelič
Jarmila Škrinárová
Zoltán Geler
Miroslav Michalko
Michal Štepanovský
Ján Genči
Grzegorz Michalski
Katarína Teplická
Andrzej Grosser
Mykola Nikitchenko
Martin Tomášek
Andrzej Grzybowski
Valerie Novitzká
Michal Vagač
Zdeněk Havlice
Ľuboš Ovseník
Ján Vaščák
Dana Horváthová
Volodymyr Ovsyak
Valentino Vranić
Ján Hurtuk
Rastislav Pištej
František Zbořil
Sergej Chodarev
Dijana Plantak-Vukovac
Jaroslav Zendulka
Ondrej Kainz
Horia F. Pop
Doina Zmaranda
András Keszthelyi
Zoltán Porkoláb
Valentina Kirinić
Herbert Prähofer

TABLE OF CONTENTS

Invited Papers

Regular Papers

On some recent advancements within the pairwise comparison methodology

Andrzej Z. Grzybowski
Institute of Mathematics
Czestochowa University of Technology
Czestochowa, Poland
andrzej.grzybowski@im.pcz.pl

***Abstract*— This paper is devoted to the inconsistency analysis within the pairwise-comparisons-based inference. We show that some conventional beliefs demonstrated in literature can be questioned. New important discoveries related to this issue were reported in recent literature. These new results indicate the necessity of the development of new methods for the inconsistency analysis. Here we also present some examples supporting this thesis. We also emphasize the need for profound simulation-studies with the aim of developing new methodology for the evaluation of the quality of pairwise comparison matrices as sources of information for priorities estimation.**

Keywords—multi-criteria analysis; pairwise comparisons; AHP; prioritization; consistency index

I. Introduction

In real-world problems every decision is made within a decision environment determined by the collection of decision-alternatives, the knowledge about consequences of their specific choices and preferences connected with the final results. From the normative theory standpoint, a decision problem is well-defined if a proper preference relation is determined on the set of possible outcomes. If such preference relation satisfies requirements, that are called preference axioms, then it is a straightforward optimization problem where a number of optimization methods can be applied dependently on the specific framework determining the underlying model. However, there are also numerous decision problems where a *decision maker* (DM) is faced with a number of alternatives that have to be evaluated with respect to various conflicting criteria. Such problems are subjects of the theory of multi-criteria decision making.

Multi-criteria decision making consists mostly of two branches, multiple criteria optimization and multi-criteria decision analysis. The main distinction between the two groups of methods is based on the number of alternatives under evaluation. Multiple criteria optimization is concerned with problems with infinite number of alternatives, while multi-criteria decision analysis typically deals with multiple criteria problems that have a small number of alternatives often in an environment of uncertainty. Among the multi-criteria decision analysis methods perhaps the most popular is the AHP - the Analytical Hierarchy Process, see [1, 2]. This method enables decision maker to assign global priority weight to each decision alternative based on the analysis of different conflicting criteria and their relative importance. AHP was originally developed by Saaty [1], and since then it has been an effective tool in modeling situations involving multiple conflicting objectives. The main purpose of using AHP is to identify the preferred alternative and also determine a ranking of all alternatives when the decision criteria are considered simultaneously. In real world problems AHP is applied by breaking down an unstructured problem into component parts.

Nowadays the pairwise comparison is a common technique that is primarily used in the *analytic hierarchy process* (AHP) - one of the most popular tools for *multi-criteria decision making* (MCDM). The AHP was developed in the seventies and eighties of the last century by Thomas Saaty. From the very beginning Saaty's concept gained much attention of DMs' community and had a great impact on the development of the pairwise comparisons based prioritization methodology. Contemporary applications of the AHP as well as recent trends in the theoretical analysis within this decision-making framework are described e.g. in [3,4].

In the AHP practice the DM is asked to make pairwise comparison judgments to express relative strength in which a given alternative is better/worse than the other one. They opinion is often expressed in linguistic terms and then transformed to numbers chosen from an adopted set that is called "a scale". The core of the AHP is the pairwise-comparisons based estimation of priority weights.

Two main issues related to pairwise comparisons are of special practical and formal interest: the prioritization methods and the consistency measurement. In this paper we focus on the latter. We show that some conventional beliefs demonstrated in literature are not well-justified. We also argue that the development of some new methodology that can be adopted for the pairwise comparison matrix acceptance is necessary.

II. The Prioritization Problem

Let us consider a problem of ranking n decision alternatives. A problem of prioritization (or deriving priority weights) is to estimate a *priority vector* (PV) $\boldsymbol{w} = (w_1, \dots, w_n)$ on the base of a matrix $\boldsymbol{A} = [a_{ij}]_{n\times n}$, with the elements a_{ij} being the DM judgments about the priority ratios w_i/w_j , $i, j = 1, \dots n$. Such a matrix $\boldsymbol{A}$ is called a *pairwise comparisons matrix* (PCM).

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Many prioritization methods have been proposed in literature. In [5] the authors compared 18 methods which may be used for this purpose. Their list of methods was corrected in [6]. After these publications yet another interesting proposals have been presented in various paper, e.g. [7, 8]. The variety of methods result from various concepts of the model of uncertainty, different estimation criteria and different assumptions about the judgments' errors. Among practitioners the most popular ones are the *right eigenvalue method* (REV) introduced by Saaty [1, 2] and the *geometric mean method* (GM) developed by Crawford and Williams, [10].

The idea of the REV can be described very briefly in the following way. In a perfect judgment case, where the PCM is a consistent matrix the following equation holds:

$$\boldsymbol{A}\boldsymbol{w} = n\boldsymbol{w} \tag{1}$$

Thus in this case the PV $\boldsymbol{w}$ can be calculated by solving the above eigenvector equation. It turns out that for a consistent matrix $\boldsymbol{A}$ the number n is the principal eigenvalue of $\boldsymbol{A}$. It is also the only nonzero eigenvalue in this case. If the matrix $\boldsymbol{A}$ is not consistent but still reciprocal, the Saaty's proposal is to use the normalized *right eigenvector* associated with the largest eigenvalue as an estimate of the true priority vector. Hence to obtain the estimate we solve general eigenvector equation:

$$\boldsymbol{A}\boldsymbol{w} = \lambda_{\max}\boldsymbol{w} \tag{2}$$

where $\lambda_{\max}$ is the principal eigenvalue. For an arbitrary positive reciprocal matrix $\boldsymbol{A}$ the value $\lambda_{\max}$ is always real, unique and not smaller than n.

Among other prioritization methods the most popular is GM. Imposing the normalization condition $\sum_{i=1}^{n} w_i = 1$, the estimates of weights in GM can be obtained from the following formula:

$$w_i = \left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}} / \sum_{i=1}^{n}\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}} \tag{3}$$

Apart from deriving priority vectors, the second important problem is the inconsistency measurement. It is well understood that significant errors in the PCM may make the information contained in its elements completely useless.

III. The Issue Of Inconsistency In Pairwise Comparisons

As we indicated herein before one of the two important parts of the MDA under the framework of the AHP is the inconsistency measurement. It is claimed (and it seems to be natural) that serious inconsistencies in judgments about the priority ratios make the data contained in PCM useless and that they may result in poor PV estimates, see e.g. [1]. In decision making practice it is a very important problem. Therefore, in recent years, we are presented with a number of papers dealing solely with the analysis of the inconsistency of the PCM, e.g. [9, 11, 12, 13, 14, 15, 16,17,18]. Such analysis is considered as a very important topic because the possibility of evaluating the inconsistency of decision makers is of special importance in the AHP due to its possible impact on the PV-estimation errors. To improve the consistency of the PCM at hand, a number of procedures enabling the consistency "improvement" and/or "controlling" was proposed, e.g. [14, 16, 19, 20,21].

Now let us address the inconsistency problem in more details. Formally, we have two notions that are important in this context.

Definition 1

A given PCM is said to be *reciprocal* if $a_{ij} = 1/a_{ji}$.

Definition 2

A PCM is called *consistent* if it is reciprocal and its elements satisfy the condition:

$$a_{ij}a_{jk} = a_{ik} \tag{4}$$

for all $i, j, k = 1, \dots, n$.

It is obvious that in reality it cannot be expected that the elements of the PCM give true priority ratios. The evaluations of the ratios may depend on personal taste, experience, specific knowledge, etc. One cannot also neglect rounding errors if we use discrete numeric scale. Therefore typically, even if the comparisons are done very carefully, the PCM is inconsistent.

In the pairwise comparisons methodology the term "inconsistency" is treated as a synonymous to "deviation of the PCM from the consistent case as stated in Definition 2". In order to "measure" such understood inconsistency of a given PCM, various characteristics (called indices) are proposed. All the inconsistency indices share common features: they are nonnegative and they equal 0 only in the case of a perfectly *consistent* PCM.

The users of these indices also *hope* that greater index values indicate worse consistency of the DM *judgments*. However such a claim is supported only by some heuristic arguments. Among the literature we can find some interesting attempts to construct a system of "psychologically-justified" axioms which should be satisfied by "good" inconsistency indices [15, 18].

Another claim, fundamental for many applications, is that less consistent the DM judgments result in poorer PV estimates. It seems to be true, but it turns out that it is not always - below we provide example showing that the *improving* of the DM judgments *consistency* may lead to the *increment of* PV-estimate's *errors*.

So, we have two problems expressed in literature. The first one: to choose a measure of DM's judgments inconsistency, and the second one: to choose a measure of the usefulness of the PCM as a source of information about the priority vector PV. However, as it was argued in [22], in spite of very popular belief, these two problems are neither the same nor equivalent. Thus here we focus on the study the relationship between the PCM *inconsistency* and the PV-estimate's *errors,* our primal interest in the multi-criteria-decision-analysis.

IV. The Inconsistency Indices

We have two most popular proposals for inconsistency indices. One widely accepted is due to Saaty [1] and is closely related to REV. According to this concept the inconsistency of the data is characterized by the so-called consistency index SI that is computed according the following formula, [1, 2]:

$$\mathrm{SI}(\boldsymbol{A}) = (\lambda_{\max} - n)/(n-1) \tag{5}$$

where $\lambda_{\max}$ is given in (2).

Another inconsistency index is related to GM. It was proposed in [10]. According to this approach the inconsistency index is given by the formula:

$$\mathrm{GI}(\boldsymbol{A}) = \frac{2}{(n-1)(n-2)} \sum_{i<j} \log^2(a_{ij} w_j / w_i) \tag{6}$$

Apart from the two above we have also a third inconsistency index which received attention in the literature. This characteristic is due to Koczkodaj, see e.g. [17]. To define this index we need the notion of a triad. Let $\boldsymbol{A} = [a_{ij}]_{n\times n}$ be a given PCM. For any three different decision alternatives we have three meaningful priority ratios, say α,β,χ. Following the Koczkodaj's definition we will call the tuple (α,β,χ) a *triad* if $\alpha = a_{ik}$, $\beta = a_{ij}$, $\chi = a_{kj}$ for some different $i, j, k \le n$. The *triad inconsistency* is defined by the formula:

$$\mathrm{TI}(\alpha,\beta,\chi) = \min\left[\left|1-\frac{\beta}{\alpha\chi}\right|, \left|1-\frac{\alpha\chi}{\beta}\right|\right] \tag{7}$$

The inconsistency index of a given PCM is defined as a maximum of all triad inconsistencies e.i.

$$\mathrm{KI}(\boldsymbol{A}) = \max\,[\mathrm{TI}(\alpha,\beta,\chi)] \tag{8}$$

where the maximum is taken over all possible triads in the upper triangle of the PCM.

Yet another index , introduced recently, is also based upon Koczkodaj's idea. Koczkodaj's index KI, as defined as the *maximum* of all triad inconsistencies, is rather robust against the changes in the number of errors if their magnitude is the same. Thus it was proposed in [22] to use more sensitive characteristic: the average value of all "triad inconsistencies". So, the new index ATI is defined by the formula:

$$\mathrm{ATI}(\boldsymbol{A}) = \mathrm{Mean}\,[\mathrm{TI}(\alpha,\beta,\chi)] \tag{9}$$

where the above arithmetic mean is computed on the basis of all different triads (α,β,χ) in the upper triangle of the considered PCM.

At this point we should note that – from mathematical point of view – we have possibly infinitely many nonnegative matrix functions that can "measure" the deviation of any given PCM from the consistent one, and be equal to 0 in the perfect case. For instance, one may consider indices constructed similarly as KI or ATI but with different triad-inconsistency definition, e.g. we may consider:

$$\mathrm{TI2}(\alpha,\beta,\chi) = \frac{|\alpha\chi-\beta|}{\alpha\chi+\beta} \tag{10}$$

or

$$\mathrm{TI3}(\alpha,\beta,\chi) = \left(\frac{\alpha\chi-\beta}{\alpha\chi+\beta}\right)^2 \tag{11}$$

as new triad inconsistency characteristics.

V. Improving Consistency Versus Estimation Quality – Examples

Let us start with the following example - a kind of a mental experiment.

Example. Let us consider a problem where a DM needs to rank three decision alternatives. Let us assume that *her/his true* PV is known and equal to $\boldsymbol{v} = (0.5, 0.32, 0.18)$. The *matrix of the true priority ratios* (MPR) related to this vector $\boldsymbol{v}$ is the following:

$$\mathrm{MPR}(\boldsymbol{v}) = \begin{bmatrix} 1 & 25/16 & 25/9 \\ 16/25 & 1 & 16/9 \\ 9/25 & 9/16 & 1 \end{bmatrix}$$

Let us also assume that our DM follows the most popular in the AHP approach and he/she uses Saaty's scale that contains the integers from 1 to 9 and their reciprocals. If our DM is very trustworthy, then the following PCM is produced:

$$\boldsymbol{M1} = \begin{bmatrix} 1 & 2 & 3 \\ 1/2 & 1 & 2 \\ 1/3 & 1/2 & 1 \end{bmatrix}$$

Let us note that this PCM is not consistent, and the only reason for this is the adoption of the suggested by the AHP procedure. Now let us try to make the PCM consistent. As suggested in literature it would be the best case, and it is the desired situation which we want to obtain with the help of various algorithms designed for consistency improvement. In our simple case it can be easily done by changing the third element in the first row. We obtain the following matrix:

$$\boldsymbol{M2} = \begin{bmatrix} 1 & 2 & 4 \\ 1/2 & 1 & 2 \\ 1/4 & 1/2 & 1 \end{bmatrix}$$

Now our matrix is consistent, however apparently it is not error-free. But all inconsistency indices take on the value 0 and it suggests that we deal with an ideal situation, but it is not so - based on such a PCM we make estimation errors. Indeed, in such a case every prioritization method gives the same PV estimate, here: $\boldsymbol{w2} = (4/7, 2/7, 1/7)$. Let us compute the estimate errors:

- average absolute error :

$$\mathrm{AE}(\boldsymbol{v}, \boldsymbol{w2}) = \frac{1}{N}\sum_{i=1}^{N} |v_i - w_i| \cong 0.476$$

- average relative error:

$$\mathrm{RE}(\boldsymbol{v}, \boldsymbol{w2}) = \frac{1}{N}\sum_{i=1}^{N}\frac{|v_i - w_i|}{v_i} \cong 15\%$$

It is interesting to compare these errors with the errors associated with the estimates received on the basis of the inconsistent matrix $\boldsymbol{M1}$. In this case we have the following PV estimate calculated with the help of REV: $\boldsymbol{w1} = [0.539615,\ 0.296961,\ 0.163424]$, and the errors are:

- absolute: $\mathrm{AE}(\boldsymbol{v}, \boldsymbol{w1}) = 0.26$
- relative: $\mathrm{RE}(\boldsymbol{v}, \boldsymbol{w1}) = 8\%$

These errors are almost twice smaller than in the previous case, where the matrix was "perfectly improved"!

So, we have made interesting observations studying this very simple example. First, the consistent matrix and the error-free matrix is not the same - an important observation that was discussed also e.g. in [22] and [23]. Second: the improvement of PCM consistency may increase the PV estimation errors!

As we mentioned in Introduction, the principal aim for which the inconsistency indices are developed – at least within the MCDM - is to indicate the PV-estimation quality. The above example shows that a "poor", inconsistent matrix allows one to derive significantly better estimates than the ones received on the basis of an improved and consistent one. But, as it was emphasized in [22], the *right question* here is: ***how likely is it?*** Therefore, to find a good estimation-quality-indicator, we should state problem in a statistical manner: given a fixed value of an inconsistency index ***how probable it is to*** obtain good estimates of PV and/or *how often* in the same situation the estimates appear to be poor? Another important problem is the following: which inconsistency index is the best one in giving answers for such kind of questions?

So, in order to find a proper inconsistency characteristic we need to make statistical studies. However, the only way one can do it, is via computer simulations.

VI. Simulation Experiments In The Studies Of The Indices Performance

Monte Carlo simulations have been used in the studies of prioritization methods since the very beginning of the applications of the AHP, see e.g. [24]. Quite recently a common simulation framework was proposed (see [5, 6]) to compare the performance of various prioritization methods. This framework was significantly modified in [8, 22] in order to receive observations that can form a base for more general conclusions about prioritization methods. Moreover, this framework was also adopted, for the first time in literature, to study the indices performance as possible indicators of the PV-estimates quality. Those simulation experiments reveal new and quite surprising facts. For example let us look at Table 1. It presents the results of simulation studies described in [22]. In order to save the article space we describe it shortly, for details see [22]. Within the simulation framework the "true" PV was randomly generated, then the perfect PCM was randomly disturbed (to simulate the real behavior of the DM). Next both the PV estimates and the related values of the considered inconsistency indices were computed at the same. In case of good inconsistency index it should be expected that the increment of its value is closely related to the increment of the PV-estimation errors. Thus, in the analysis of inconsistency indices the most important is the Spearman rank correlation coefficient ρ between the indices values and all kind of location parameters describing the error-distribution (such as mean, median or quantiles). In the case of well-performing index in each case ρ should be close to 1. Obviously, its value may depend on the prioritization method. Thus we take into account both most popular ones described in Section 2. The symbols AE(REV), AE(GM) stand for the absolute errors computed in cases where the PV-estimates are received with the help of indicated prioritization method.

Table 1 shows example of Spearman rank correlation coefficients for considered here indices. We use the word "example" because the values may be different dependently on the simulation framework (e.g. the assumed probability distribution of the random factors, the presence of so-called big errors, the number of alternatives) and the considered statistical location parameters. However in various frameworks considered in [22] the values of ρ are very similar. The results were also very much the same in case of the other consider type of error, i.e. the RE(REV), RE(GM). In each case they indicate that both SI and GI are worse than KI and ATI, and the latter is the best index from the considered here point of view.

TABLE I. Average Spearman rank correlation coefficients between the mean values of an inconsistency index computed in its classes IC$_l$ (l=1,...,15) and the indicated statistical characteristics of PV-estimation errors corresponding to these classes. The presented results are computed in the case of N=4 on the basis of 240 000 PCM-s.

Stat. charact.		**SI**	**GI**	**KI**	**ATI**
		***Correlations with the errors* AE(REV)** [a]			
Mean		0.536	0.525	0.921	1
Quantiles	p = 0.1	0.729	0.754	0.961	1
	Median	0.729	0.743	0.964	1
	p=0.9	0.421	0.411	0.871	0.868
		***Correlations with the errors* AE(GM)** [a]			
Mean		0.754	0.771	0.982	1
Quantiles	p = 0.1	0.796	0.821	0.964	1
	Median	0.793	0.854	0.979	1
	p=0.9	0.496	0.500	0.939	0.996

[a] Source: A.Z. Grzybowski, "New results on inconsistency indices and their relationship with the quality of priority vector estimation," Expert Systems With Applications 43, 197–212.

Results received via our simulations show that the SI is not such as good indicator of the possible estimates-errors as it is often assumed to be, and consequently, the inference made upon observations of its values may be rather misleading. This fact even better can be seen in Fig. 1. The graphs presented in it are based on the data from [22], where the reader is referred for more profound discussion of this issue. Let us also note, that similar studies can be performed for new indices, such as the examples presented at the end of Section 3. It is possible that they appear to be better than the already known ones. Preliminary

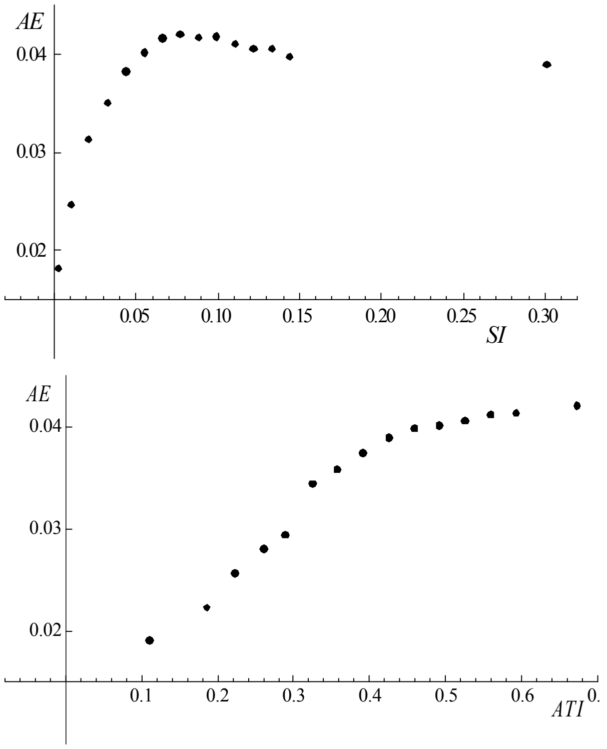


Fig. 1. *Performance of the indices SI and ATI.* The relation between mean values of SI (upper graph) and ATI (lower graph) in successive classes of their values vs. mean values of the errors AE in corresponding classes of PV-estimates.

simulations that we have already conducted gave quite promising results.

VII. Final Remarks

In the light of the facts presented in Sections 3 and 4, there is a possibility to upgrade the Analytic Hierarchy Process and improve the theory of choice. Further studies should focus on description and characterization of fundamental concepts of different types of inconsistency and their indicators. Another important task is the sensitivity analysis: i.e. the analysis of the relationship between the type/number/magnitude of judgment errors and the final (aggregated) rankings perhaps modelled in terms of its distribution in the probabilistic framework. As a result, the future research should certainly involve elaboration of a new method for consistency measurement along with a new procedure for the PCM acceptance. Such a procedure should rely on sound fundamentals developed within the statistical inference theory. In our opinion, a possibility exists to design and elaborate an innovative methodology for inconsistency characterization, which potentially should be significantly more reliable than the methodology based on the analysis of SI or GI.

References

[1] T.L. Saaty, "The analytic hierarchy process," McGraw Hill, New York 1980. ISBN-13: 978-0070543713.

[2] T.L. Saaty, "Decision making with the AHP: Why is the principal eigenvector necessary," Euro. J. Oper. Res. 2003, 145, 85-91.

[3] Emrouznejad & M. Marra, "The state of the art development of AHP (1979–2017): a literature review with a social network analysis", International Journal of Production Research, 2017, DOI: 10.1080/00207543.2017.1334976.

[4] G. Kou, D., Ergu, C., Lin, C., & Y. Chen, " Pairwise comparison matrix in multiple criteria decision making", Technological and Eco-nomic Development of Economy, 2016, 22(5), 738-765.

[5] E.U. Choo and W.C. Wedley, "A common framework for deriving preference values from pairwise comparison matrices," Comp. Oper. Res. 2004, 31, 893–908.

[6] C.-C. Lin, "A revised framework for deriving preference values from pairwise comparison matrices," European Journal of Operational Research 2007, 176, pp.1145–1150.

[7] R.K. Goyal and S. Kaushal, "A constrained non-linear optimization model for fuzzy pairwise comparison matrices using teaching learning based optimization," Applied Intelligence 2016, 45(3), 652-661.

[8] A.Z. Grzybowski, "Note on a new optimization based approach for estimating priority weights and related consistency index," Expert Systems with Applications 2012, 39 , pp.11699-11708.

[9] P. Kazibudzki, "Redefinition of triad's inconsistency and its impact on the consistency measurement of pairwise comparison matrix," Journal of Applied Mathematics and Computational Mechanics 2016, vol. 15, no. 1, pp. 71-78.

[10] G. Crawford and C. Williams, "A note on the analysis of subjective judgment matrices," J. Math. Psychol. 1985, 29:387–405.

[11] J. Aguarón and J.M. Moreno-Jiménez, "The geometric consistency index: approximate thresholds," European Journal of Operational Research 2003, 147 (1), pp. 137-145.

[12] J. Aguarón, M.T. Escobar and J.M. Moreno-Jiménez, "The precise consistency consensus matrix in a local AHP-group decision making context," Annals of Operational Research 2014, DOI 10.1007/s10479-014-1576-8.

[13] J. Alonso and T. Lamata, "Consistency in the analytic hierarchy process: A new approach," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2006, 14, pp.445–459.

[14] J. Benítez, X. Delgado-Galván, J. Izquierdo and R. Pérez-García, "Improving consistency in AHP decision-making processes," Applied Mathematics and Computation 2012, 219, 2432–2441.

[15] M. Brunelli and M. Fedrizzi, "Axiomatic properties of inconsistency indices for pairwise comparisons," Journal Of The Operational Research Society, doi:10.1057/jors.2013.135.

[16] S. Bozóki and T. Rapcsák, "On Saaty's and Koczkodaj's inconsistencies of pairwise comparison matrices," J. Glob. Optim. 2008, 42(2):157–175

[17] W.W. Koczkodaj, "A new definition of consistency of pairwise comparisons," Math. Comput. Model. 1993, 18(7):79–84

[18] W.W. Koczkodaj and R. Szwarc, "On axiomatization of inconsistency indicators for pairwise comparisons," Fundamenta Informaticae 2014, 132(4), 485-500.

[19] W.W. Koczkodaj and S.J. Szarek, "On distance-based inconsistency reduction algorithms for pairwise comparisons," Logic Journal of the IGPL 2010, 18(6), 859-869.

[20] K. Kułakowski, R. Juszczyk, S. Ernst, „A concurrent inconsistency reduction algorithm for the pairwise comparisons method", Artificial Intelligence and Soft Computing 2015, 9120, 214–222.

[21] M. Lamata and P. Peláez, "A method for improving the consistency of judgements," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 2002, 10 (6) 677-686.

[22] A.Z. Grzybowski, "New results on inconsistency indices and their relationship with the quality of priority vector estimation," Expert Systems With Applications 2016, 43, 197–212, ISSN: 0957-4174.

[23] J. Temesi, "Pairwise comparison matrices and the error-free property of the decision maker," Central European Journal of Operational Research 2011, v.19, pp.239–249.

[24] F.A. Zahedi, "Simulation study of estimation methods in the analytic hierarchy process," Socio-Econ. Plann. Sci. 1986, 20, 347–354.

How to Apply Model-Driven Paradigm in Information System (Re)Engineering

Sonja Ristić
University of Novi Sad
Faculty of Technical Sciences
Novi Sad, Serbia
sdristic@uns.ac.rs

***Abstract*—Inability of traditional information system development approaches, methodologies and tools to cope with ever increasing complexity of contemporary information systems leads towards paradigm shift. Model-driven (MD) paradigm assumes orientation on models at all stages of system development. MD approach to information system and software engineering addresses complexity through abstraction. The approach promotes the idea of abstracting implementation details by focusing on: models as first class entities and automated generation of models or code from other models. The role of model-driven paradigm in information system (re)engineering is illustrated on the example of IIS*Studio development environment aimed at MD information system (re)engineering.**

Keywords—model-driven paradigm, information system engineering, reengineering, domain specific modeling languages

I. Introduction

An information system (IS) serves the different concerns of numerous stakeholders. Its design and implementation involve understanding of the social and organizational context of the system and making decisions according to the limitations of environment and technology. The open, dynamic and almost unbounded nature of contemporary environment of the ISs poses many new challenges. The organization of contemporary enterprises is network oriented and spanned across organizational boundaries. Information technology evolution from client-server architectures and service oriented architectures that characterized data and process-centric ISs, to semantic networks, social media, Internet of Things and cloud computing influences significant changes of ISs. The traditional information systems development methodologies, approaches and tools would be replaced with those that can support and improve the business relevance, ubiquitous, flexibility, scalability, interoperability and integration of ISs in the turbulent business environment. ISs are becoming increasingly complex due to the both essential and accidental complexity [1]. Essential complexity, inherent to the problem being solved, cannot be mitigated. On the other side, accidental complexity is caused by inappropriate selection or usage of IS's development approaches, methodologies, techniques or tools. Deploying traditional IS design methodologies requires advanced knowledge, skills, and high perception power of IS designers and developers. According to [2] and [3], these methods and techniques are often incomprehensible to end-users. In practice, that may lead to problems in communication and to misunderstanding between designers and end-users. Consequently, they do not provide conditions to fully incorporate end-users in IS design and implementation process. End-users are practically excluded from the design process after they pose and initially discuss their requirements with a design team. Further consultations with them are too infrequent and too abstract to result in any meaningful engagement until the system is implemented. The implementation process remains a manual task that is error-prone and time-consuming. It is still very hard to uncover the problems, and to provide the opportunity to correct mistakes early in the development process. The paradigm shift is needed to overcome these disadvantages of traditional IS's development approaches. In the paper we deal with an approach based on model-driven paradigm.

Model-driven paradigm assumes orientation on models at all stages of system development. A complex system consists of several interrelated models organized through different levels of abstraction and platform specificity. The creation of models is the result of a modeling process that can be supported by a modeling tool. The modeling process is aimed at understanding of: the system's context, the collaboration required to achieve the goals of the system, the distribution of system functions and the set of its constraints. Each model is expressed by the concepts of a modeling language. Meta-models are used to define the modeling languages. In that way, a meta-model of a modeling language defines a set of valid models expressed by the concepts of that language. These models are said to be conformant to the meta-model. Models make it possible to hide irrelevant details, provide different model viewpoints, and isolate and modularize models of cross-cutting concerns of a System under Study (SUS). In the context of IS engineering Model Driven Engineering (MDE) and Model-Driven Software Engineering (MDSE) are not just a model-driven approaches to IS and software engineering development. They are usually centered around the languages that are specific to a certain domain of application—Domain Specific Language (DSL) and Domain Specific Modeling Languages (DMSL). The models in MDSE need to be refined, integrated and used to produce code and therefore they would undergo a series of transformations. Each transformation adds levels of specificity and detail. A chain of model-to-model (M2M) transformations is completed starting from an initial model at the highest level of abstraction (Platform Independent

Supported by Ministry of Education, Science and Technological Development of Republic of Serbia, Grant III-44010.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Model, PIM), through the less abstract models, with different levels of platform specificity (Platform Specific Models, PSMs), and resulting in an executable program code that represents a model at the lowest level of abstraction (fully PSM). These M2M transformations transform a model conformant to a meta-model into another one conformant to a different meta-model. MDE and MDSE combine DSMLs with corresponding transformation engines and generators to alleviate IS complexity and express domain concepts effectively [2].

In order to illustrate the role of model-driven paradigm in IS (re)engineering we present IIS*Studio development environment (IIS*Studio DE), that is developed through a number of research projects on model-driven intelligent systems for information system development, maintenance and evolution. It is aimed to provide the IS design, generating executable application prototypes and IS reverse engineering. Our approach is mainly based on the model-driven IS and software engineering [4]–[11] and DSL paradigms [3], [12] [13]. The main idea was to provide the necessary PIM meta-level concepts to IS designers, so that they can easily model semantics in an application domain. Afterward, a number of formal methods and complex algorithms may be utilized to produce database schema specifications and IS executable code, without considerable expert knowledge.

Apart from Introduction and Conclusion the paper has three sections. In Section 2 a brief history of IIS*Case development is presented, with focus on motivation for paradigm shift. Architecture of IIS*Studio DE is described in Section 3, and main steps of model-driven database reengineering process are given in Section 4.

II. Motivation for paradigm shift

In early stages of our research in late 1980s and early 1990s the first version of IIS*Case tool was developed. It was aimed at: (i) conceptual modeling of a database (db) schema, (ii) automated design of relational database subschemas in the 3rd normal form, and (iii) automated integration of subschemas into a relational db schema. It is based on a methodology of gradual integration of independently designed subschemas into a database schema. Our motivation to develop IIS*Case was three folded.

- Most of CASE (Computer Aided Software Engineering) tools at that time have used ER (Entity-Relationship) data model to express conceptual database schema. In contrast, in our approach that is supported by IIS*Case, conceptual schemas are expressed by sets of the form types. The main motivation of introducing form type is using a concept that is more familiar to end-users' perception and that is formal enough to precisely express all the rules significant for structuring future db schema [14]. In the way we have laid the foundations of future DSML to be used in actual version of IIS*Studio that contains IIS*Case as a component.
- Implementation database schema, often expressed in relational data model, in majority of CASE tools was obtained by applying transformation rules onto a previously created ER database schema. Unfortunately, such a process could not guaranty gathering a relational db schema in 3rd normal form. That was a motive to use a well-known Beeri & Bernstein synthesis algorithm [15] to overcome the aforementioned problem. The synthesis algorithm is improved so as to process not only functional dependencies and generate relation scheme keys only, but also other kinds of relational database constraints [16], [17]. It was the first transformation tool that we have deployed in IIS*Case tool to provide automatic generation of relational database schema specifications from the conceptual database schema model expressed by form types.
- Complexity of user requests is very often beyond the designer's power of perception. That was our motive to support independent and gradual database schema design. We use the notion of external schema that is a structure at the conceptual level which formally specifies a user view on a db conceptual schema. Design of a complex db schema is based on a gradual integration of separately designed external schemas. After being created, external schemas are integrated into a conceptual db schema. Unfortunately, it is difficult and sometimes even impossible to formalize the process of integration of the external schemas expressed by means of ER or from type data model. However, the same problem of integration could be solved easier in relational data model. According to Bezivin [6] the technical space may be defined as: “a working context with a set of associated concepts, body of knowledge, tools, required skills, and possibilities”. So, we can say that form type data model and relational data model belong to different technical spaces. In, so called, relational technical space there are mechanisms that easy the process of subschema integration. Therefore, we have decided to change technical space in order to solve the problem of external schema integration. The designed external schemes are first transformed into relational subschemas, and afterwards the synthesis algorithm is used to generate integrated db schema [18], [19]. In the way we have used the idea of using different technical spaces; specifying, generating and executing M2M transformations; and using (at that time) three-layered model hierarchy that could adhere to the common meta-meta-model Meta-Object Facility (MOF) emerged a few years later as an OMG standard [20].

In that way, the roots of our IIS*Case meta-model were created and the pathway of the future paradigm shift towards the IIS*Studio DE based on model-driven paradigm was prepared in early 1990s. From that time IIS*Case meta-model is continuously developing. Till the mid of 2000s, we worked on continuous improvements of the synthesis algorithm and its implementation into the IIS*Case tool [17]. In that way we have improved the process of system integration that is not just a mere unifying of its subsystems. It is based on detecting and resolving all the formal constraint collisions. In [21], [22] we have proved that, at the level of relational data model, it is possible to automatically detect formal collisions of database constraints embedded into different subschemas, where each subschema represents a database schema of a sole IS subsystem. In [19] for the first time we explicitly specify our form type model as PIM and extend it with the concepts related with the business application specification. In [23] and [24] we have presented features of SQL Generator that is implemented into IIS*Case, alongside with selected methods for implementation of database constraints, using mechanisms provided by a relational DBMS. A PIM model of business applications is combined with a selected common UI model—that is a PIM, too—and then automatically is transformed into the program code [25]. Further improvements are made by modeling check constraints and untypical functionalities of business applications [26]. All presented functionalities are steps of forward engineering process. In order to adhere to MD paradigm and to ensure model synchronization, reverse engineering of relational databases to conceptual data models is implemented, too. Different aspects of implemented reverse engineering process are presented in [27]–[34].

One of the main goals of IIS*Studio is to provide conceptual modeling by creating platform independent models based on form type (FT) data model. As designers can use some other approaches, we have decided to extend the IIS*Studio functionalities in order to support conceptual modeling based on Extended ER (EER) data model. The tool is named Multi-paradigm Information System modeling Tool (MIST) [35]–[37]. In MIST, both EER and FT data models may be used simultaneously, as conceptual modeling languages.

III. Architecture of IIS*Studio DE

IIS*Studio DE currently comprises four tools: IIS*Case, IIS*UIModeler, IIS*Ree and MIST. These tools communicate by means of shared repository aimed at storing project specifications (domains, attributes, form types, application systems), business application specifications, formal database specifications and implementation specifications of database subchemas and integrated database schema.

The IIS*UIModeler is an integrated part of the IIS*Studio DE, aimed at modeling of graphic user interface (GUI) static aspects. By means of IIS*UIModeler a designer specifies UI templates. The specification of the UI template is stored in the IIS*Studio repository.

In a forward engineering process, supported by IIS*Case tool, designers start with a high-level model, abstracting from all kinds of platform issues. Through a chain of M2M transformations, ending up with a model-to-text (M2T) transformation, the initial platform independent model transforms iteratively to a series of models with less degree of platform independency, introducing more and more platform specific extensions. Conversely, in a reverse engineering process, supported by IIS*Ree tool, the abstraction level of models and degree of platform independency are increasing throughout the chain of transformations.

Relational databases are at the core of most company information systems, hosting critical information for the day to day operation of the company. The knowledge captured in them can serve as an important resource in a legacy information system modernization project and they are a common source of reverse engineering processes. Starting from a physical database schema, that is recorded into the relational database schema data repository, the conceptual database schema or logical database schema may be extracted [38]. All of these db schemas represent models at different levels of abstraction. The extraction and conceptualization process may be seen as a chain of M2M transformations that trace model elements from a model at the lower level of abstraction to a model at the higher level of abstraction. The IIS*Ree tool is aimed at reverse engineering of relational databases to conceptual data models. IIS*Studio integrates functionalities of IIS*Case tool and IIS*Ree tool to provide automated support not only for the design, development and implementation of IS but also to provide automated support for the IS evolution and reengineering. In Fig. 1 we use Architecture-driven Modernization (ADM) horseshoe model [39] to illustrate main steps of reengineering approach supported by IIS*Studio. The steps of reverse engineering are presented on the left side of Fig. 1 and steps of forward engineering can be seen on the right side of Fig.1.

IV. A Model-driven approach to database reengineering process

In order to illustrate model-driven paradigm in the context of IS reengineering, here we present a model-driven approach to database reengineering process that is supported by IIS*Studio DE and that is described in [32]. In the following text we present adapted description from [32].

Reengineering generally includes some form of reverse engineering followed by restructuring (optional) and some form of forward engineering. Fig. 1 can be used to illustrate main steps of a model-driven approach to database reengineering implemented in IIS*Studio. Starting from a physical database schema, that is recorded in the relational database schema data repository, the conceptual database schema or logical database schema may be extracted. These database schemas represent models at different levels of abstraction and are conformant to different meta-models. Physical database schema is conformant with a vendor specific physical db schema meta-model, logical db schema is conformant with standard db schema meta-model or with generic relational db schema meta-model, and conceptual db schema could be conformant with form type meta-model. Database meta-models classification is explained in [31]. Reverse engineering phase covers steps 1–5, where steps 1–3 belong to the data extraction phase, and steps 4 and 5 make the data conceptualization phase. Step 6 is optional and represents restructuring phase. Forward engineering phase covers steps 7–9. Step 10 is not the step of database reengineering process and it is an activity of code reengineering process. In the first step information about supported data types, tables, columns, check constraints, primary key and unique key constraints accompanied with foreign key constraints are captured from a legacy database repository and are placed in IIS*REE repository. Additional information about the inverse referential constraints and homonym inconsistencies are discovered in the step 2.

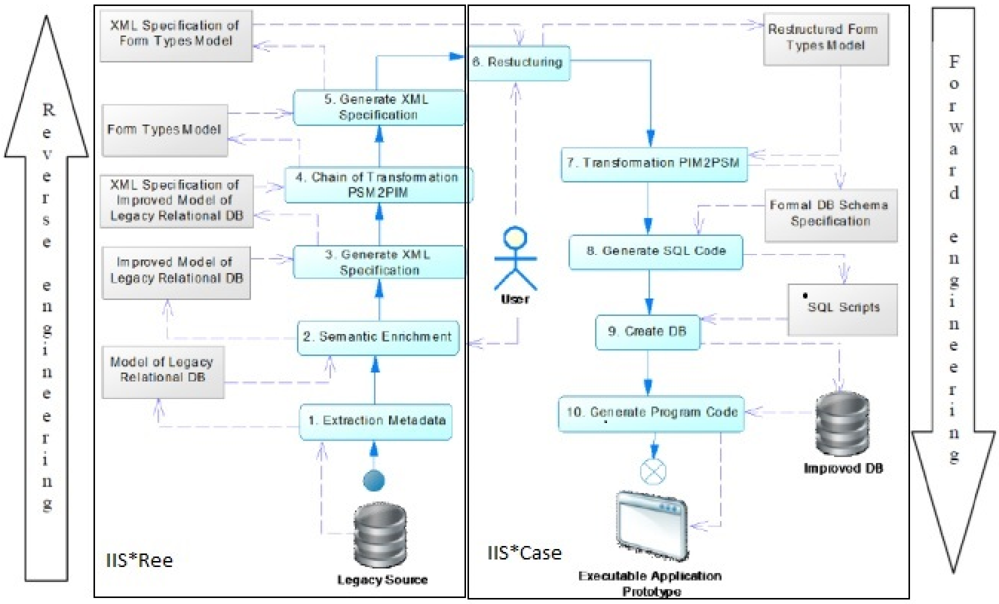

Reverse engineering
XML Specification of Form Types Model
Form Types Model
XML Specification of Improved Model of Legacy Relational DB
Improved Model of Legacy Relational DB
Model of Legacy Relational DB
5. Generate XML Specification
4. Chain of Transformation PSM2PIM
3. Generate XML Specification
2. Semantic Enrichment
1. Extraction Metadata
Legacy Source
IIS*Ree
6. Restucturing
User
Restructured Form Types Model
7. Transformation PIM2PSM
Formal DB Schema Specification
8. Generate SQL Code
SQL Scripts
9. Create DB
10. Generate Program Code
Improved DB
Executable Application Prototype
IIS*Case
Forward engineering

Fig. 1. Horseshoe model of model-driven approach to database reengineering implemented in IIS*Studio [32]

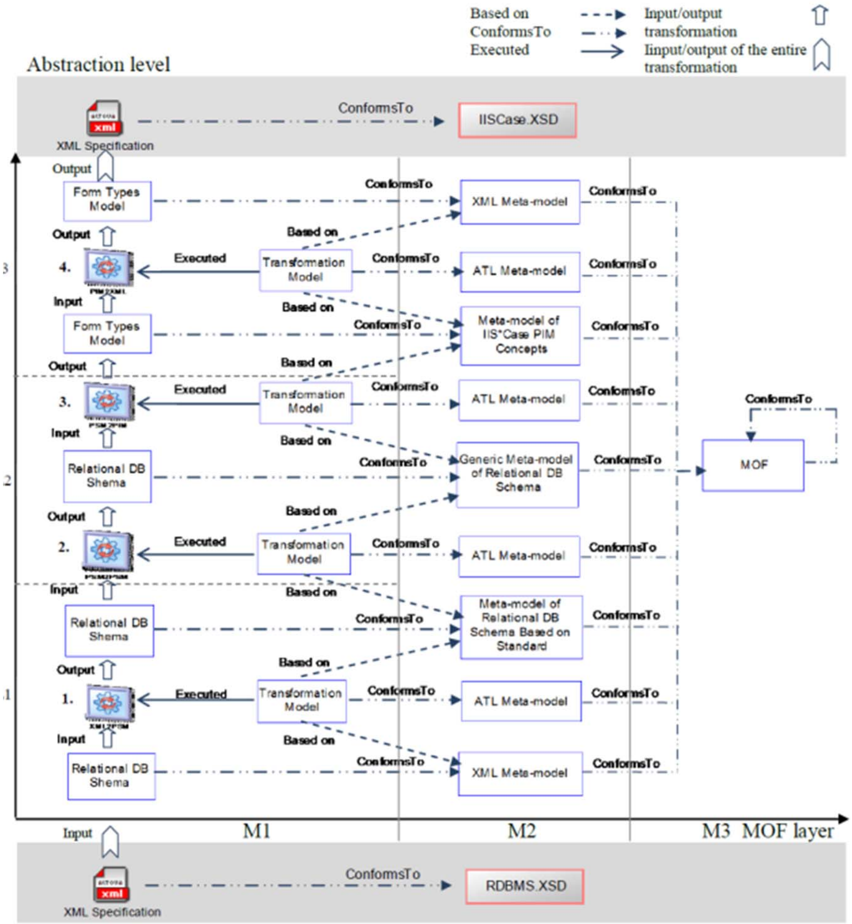


Fig. 2. Data structure conceptualization through a chain of M2M transformations [32]

Captured physical database schema may be transformed into desired conceptual database model. Depending on selected target meta-model a model transformation or a chain of model transformations will be executed. These transformations are based on appropriate meta-models that can be at different levels of abstraction. The distribution of the model transformations and supporting meta-models across MOF and abstraction levels is given in Fig. 2.

In the purpose of specifying and managing meta-models we have used the Eclipse Modeling Framework (EMF), Eclipse Juno 4.2.1. and OCL 3.2.1. Model transformations used in step 4 are implemented in ATL IDE (ATLAS Transformation Language Integrated Development Environment), version 3.3.1. Steps 1 and 2 are implemented in one technological space (OracleJ Developer), and step 4 in another one (EMF). Step 3 is used to bridge the technological gap and to export captured data by means of XML document conformant with XML meta-model. Step 6 is implemented in the same technological space as steps 1 and 2, due to the fact that it requires user interaction, too. Step 5, like the step 3, is used to bridge the technological gap, but in the opposite direction. Therefore, strictly speaking, it is not part of the data conceptualization phase.

In a forward engineering process, designers start with a high abstraction level model, abstracting from all kinds of platform issues. Through a chain of model-to-model (M2M) transformations, ending up with a model-to-text (M2T) transformation, the initial platform independent model transforms iteratively to a series of models with less degree of platform independency, introducing more and more platform specific extensions.

V. Conclusion

One of the main assumptions of the model-driven approach to information system and software development is that systems of large complexity can only be designed and maintained if the level of abstraction is considerably higher than that of programming languages. By means of models, semantics in an application domain can be precisely specified using terms and concepts the end-users are familiar with, such as the form types used in IIS*Studio are. The focus of software development is shifted from the technology domain toward the problem domain. Domain-specific models, and the modeling languages that support them, play a central role in the success of model-driven development and its ability to bridge the abstraction gap between information system and software implementation technologies and human developers.

A complex system consists of several interrelated models organized through different levels of abstraction and platform specificity. The modeling process of an IS is aimed at understanding of: the system's context, the collaboration required to achieve the goals of the system, the distribution of system functions and the set of its constraints. That was the motivation to introduce IIS*Studio PIM concepts that enable modeling of system context, collaboration, distribution of system functions and system constraints. The main idea was to actively incorporate end-user in the IS design, implementation and evolution process. Our approach to database schema generation, through the integration of independently designed external schemas, and consolidation of subschemas differentiates IIS*Studio from others tools with similar functionality. The main reason to develop our IIS*Ree reverse engineering tool was to take advantage of our approach to database schema generation during: the integration of independently designed ISs, legacy database schema restructuring and improvement of empirically designed database schemas. We continue our research and we have developed MIST. It comprises several components that support conceptual modeling based on the FT and EER approaches, as well as code generation. It enables a simultaneous usage of different approaches to the conceptual database design and IS development that may lead to the most appropriate solutions in IS design. MIST currently supports several M2M transformations: EER to relational data model (EER2Rel), EER to class data model and relational to FT data model (Rel2FT). Transformation from EER data model to FT data model is enabled by the composition of transformations EER2Rel and Rel2FT. An IS designer may select the data model that is most appropriate for the problem domain or his knowledge and preferences. In the way we confirm the claim that "In model-driven engineering, new tools are always needed" [40] and we will continue in our efforts to enable transformations between different data models.

MDE has made significant progress in addressing software engineering challenges. Some authors see MDE as first true advance in software development since the invention of compilers. The software industry is labor intensive and there is a strong need to fully automate software development. On the other side a lot of knowledge is captured in legacy ISs. Application of model-driven paradigm in both IS engineering and even more in IS reengineering could be seen as essential.

Acknowledgment

The author would like to thank to all of her colleagues she has worked with in a number of research and industry projects, and above all to Pavle Mogin, Ivan Luković, Slavica Kordić maiden Aleksić, Jelena Banović maiden Pavićević, Aleksandar Popović, Milan Čeliković and Vladimir Dimitrieski who have greatly contributed to the results presented in this paper.

References

[1] F. P. Brooks, The mythical man-month. Addison-Wesley Reading, MA, anniversary ed. edition, 1995.

[2] I. Luković, V. Ivančević, M. Čeliković, and S. Aleksić, "DSLs in action with model based approaches to information system development," In: Formal and Practical Aspects of Domain-specific Languages: Recent Developments, M. Mernik, Ed. IGI Global, Hershey, 2013, pp. 502–532.

[3] T. Kosar, N. Oliveira, M. Mernik, M. J. Varanda Pereira, M. Črepinšek, D. da Cruz, and P. R. Henriques, "Comparing general-purpose and domain-specific languages: An empirical study," Computer Science and Information Systems, vol. 7(2), pp. 247–264, 2010.

[4] C. Atkinson and T. Kuhne, "A Generalized Notion of Platforms for Model-Driven Development," In: Model-Driven Software Development.

S. Beydeda, M. Book and V. Gruhn, Eds. Berlin Heidelberg: Springer, 2005, pp. 119−136.

[5] J. Bézivin, “On the unification power of models,” Software & Systems Modeling , vol. 4 (2), pp. 171−188, 2005.

[6] J. Bézivin, “Model driven engineering: An emerging technical space,” Generative and transformational techniques in software engineering, pp. 36−64, 2006.

[7] JM. Favre, “Foundations of Model (Driven) (Reverse) Engineering: Models,”, Dagstahl Seminar Proceedings 4101, 2003.

[8] D.S. Frankel, Model Driven Architecture, Wiley, 2003.

[9] A. Kleppe, J. Warmer and W. Bast, MDA Explained, The Model Driven Architecture: Practice and Promise. Addison-Wesley, 2003.

[10] J. Mukerji and J. Miller, MDA Guide Version 1.0.1, document omg/03-06-01 (MDA Guide V1.0.1), http://www.omg.org/, Accessed October, 2017.

[11] T. Stahl, M. Völter, J. Bettin, A. Haase, and S. Helsen, Model Driven Software Development: Technology, Engineering, Management. John Wiley & Sons, Ltd, 2006.

[12] I. Dejanović, G. Milosavljević, B. Perišić and M. Tumbas, “A Domain-Specific Language for Defining Static Structure of Database Applications,” Computer Science and Information Systems, (ComSIS), vol. 7(3), pp 409–440, 2010.

[13] M. Mernik, J. Heering and A.M. Sloane, “When and how to develop domain-specific languages,” ACM computing surveys (CSUR), vol. 37(4), pp. 316–344 , 2005.

[14] P. Mogin, L. Luković, and Ž. Karadžić, “Relational database schema design and application generating using IIS*CASE tool,” In Proceedings of International Conference on Technical Informatics Timisoara, Romania: 'Politehnica' University of Timisoara, pp. 49–58, 1994.

[15] C. Beeri and P.A. Bernstein, “Computational Problems Related to the Design of Normal Form Relational Schemas,”, ACM Transactions on Database Systems, vol. 4(1), pp. 30–59, 1979.

[16] I. Luković, S. Ristić and P. Mogin, “A methodology of a database schema design using the subschemas,” In Proceedings of IEEE International Conference on Computational Cybernetics (in CD ROM). Budapest, Hungary: Budapest Polytechnic, 2003.

[17] I. Luković, P. Mogin, J. Pavićević and S. Ristić, “An approach to developing complex database schemas using form types,” Software: Practice and Experience, vol. 37(15), pp. 1621–1656, 2007.

[18] I. Luković, S. Ristić, P. Mogin and J. Pavićević, “Database schema integration process – A methodology and aspects of its applying,” Novi Sad Journal of Mathematics, vol. 36(1), pp. 115–150, 2006.

[19] I. Luković, S. Ristić, S. Aleksić and A. Popović, “An application of the MDSE principles in IIS*Case,” In Proceedings of the 3rd Workshop on Model Driven Software Engineering Berlin, Germany: TFH, University of Applied Sciences Berlin, pp. 53– 62, 2008.

[20] OMG MOF (Meta Object Facility). http://www.omg.org/mof/. Accessed October, 2017.

[21] S. Ristić, I. Luković, J. Pavićević and P. Mogin, “Resolving Database Constraint Collisions Using IIS*Case Tool,” Journal of Information and Organizational Sciences, vol. 31(1), 187–206, 2007.

[22] S. Ristić, P. Mogin and I. Luković, “Specifying database updates using a subschema,” In Proceedings of VII IEEE International Conference on Intelligent Engineering Systems Assiut-Luxor, Egypt: IEEE, Assiut University, Assiut, Egypt, and Budapest Polytechnic, Budapest, Hungary, pp. 203–212, 2003.

[23] S. Aleksić, I. Luković, P. Mogin and M. Govedarica, “A generator of SQL schema specifications,” Computer Science and Information Systems, vol. 4(2), pp. 79–98, 2007.

[24] S. Aleksić, S. Ristić and I. Luković, “An approach to generating server implementation of the inverse referential integrity constraints,” n Proceedings of the 5th International Conference on Information Technology, Amman: Al-Zaytoonah University of Jordan, pp. 1–7, 2011.

[25] S. Ristic, S. Aleksic, I. Lukovic and J. Banovic, “Form-Driven Application Development,” Acta Electrotechnica et Informatica, vol. 12(1), DOI: 10.2478/v10198-012-0002-x, FEI TUKE, Versita, Warsaw pp. 9–16, 2012.

[26] I. Luković, A. Popović, J. Mostić and S. Ristić, “A tool for modeling form type check constraints and complex functionalities of business applications,” Computer Science and Information Systems, vol: 7(2), pp. 359–385, 2010.

[27] S. Aleksić, “Methods of Database Schema Transformations in Support of the Information System Reengineering Process,” Ph.D Disertation. University of Novi Sad, Faculty of Technical Sciences, 2013.

[28] S. Aleksić, S. Ristić, I. Luković and M. Čeliković, “A Design Specification and a Server Implementation of the Inverse Referential Integrity Constraints,” Computer Science and Information Systems (ComSIS). Consortium of Faculties of Serbia and Montenegro, Belgrade, Serbia, ISSNvol. 10(1), pp. 283–320, 2013.

[29] S. Ristić, S. Aleksić, M. Čeliković and I. Luković, “Meta-modeling of inclusion dependency constraints,” In Proceedings of the 6th Balkan Conference in Informatics (BCI '13). ACM, New York, NY, USA, pp. 114–121, 2013.

[30] S. Ristić, S. Aleksić, M. Čeliković and I. Luković, “Generic and Standard Database Constraint Meta-Models,” Computer Science and Information Systems, vol. 11(2), pp. 679–696, 2014.

[31] S. Ristić, S. Aleksić, M. Čeliković, V. Dimitrieski and I. Luković, “Database reverse engineering based on meta-models,” Open Computer Science, vol. 4 (3), pp. 150–159, 2014.

[32] S. Ristić, S. Kordić, M. Čeliković, V. Dimitrieski, and I. Luković, “A Model-driven Approach to Data Structure Conceptualization,” In Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS 2015), 5th Workshop on Advances in Programming Languages (WAPL 2015), vol. 5, IEEE Computer Society Press and Polish Information Processing Society, DOI: http://dx.doi.org/10.15439/978-83-60810-66-8; 10.15439/2015F224, pp. 977–984. 2015.

[33] S. Ristić, S. Kordić, M. Čeliković, V. Dimitrieski, and I. Luković, “A model-To-model transformation of a generic relational database schema into a form type data model,” In Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdansk; Poland; DOI: 10.15439/2016F408, pp. 1577–1580, 2016.

[34] S. Kordić, S. Ristić, M. Čeliković, V. Dimitrieski, and I. Luković, “Reverse Engineering of a Generic Relational Database Schema Into a Domain-Specific Data,“ In Proceeding of the 28th Central European Conference on Information and Intelligent Systems, (CECIIS 2017), Faculty of Organization and Informatics, University of Zagreb, Varaždin, pp. 19–28. 2017.

[35] V. Dimitrieski, M. Čeliković, S. Aleksić, S. Ristić and I. Luković, “Extended Entity-Relationship Approach in a Multi-Paradigm Information System Modeling Tool,” In Proceedings of Federated Conference on Computer Science and Information Systems (4; Warsaw; 2014), IEEE Computer Society Press and Polish Information Processing Society, vol. 2, pp. 1611 – 1620. 2014.

[36] M. Čeliković, V. Dimitrieski, S. Aleksić, S. Ristić and I. Luković, “A DSL for EER Data Model Specification,” In Proceedings of International Conference on Information Systems Development (23; Varaždin; 2014), University of Zagreb, Faculty of Organization and Informatics, pp. 290–297. 2014.

[37] V. Dimitrieski, M. Čeliković, S. Aleksić, S. Ristić, A. Alargt and I. Luković, “Concepts and Evaluation of the Extended Entity-Relationship Approach to Database Design in a Multi-Paradigm Information System Modeling Tool,” Computer Languages Systems and Structures, Elsevier Inc., DOI: 10.1016/j.cl.2015.08.011, 2015.

[38] J-L. Hainaut, J. Henrard, V. Englebert, D. Roland and J-M. Hick, Database Reverse Engineering. In Encyclopedia of Database Systems, L. Liu and T. Özsu, Eds, Springer-Verlag, 2009.

[39] OMG ADM(Architecture Driven Modernization). http://www.adm.omg.org. Accessed October, 2017.

[40] J. Davies, J. Gibbons, J. Welch and E. Crichton, “Model-driven engineering of information systems: 10 years and 1000 versions,” Science of Computer Programming, Special issue on Success Stories in Model Driven Engineering, vol. 89, Part B, pp. 88–104, 2014.

Augmented Virtuality for the Next Generation Production Intelligence

Prof. Imre J. Rudas
Óbuda University, Budapest, Hungary
rudas@uni-obuda.hu

PLENARY PAPER

Abstract—**Several compelling IT trends of the past few years show us how the next possible paradigm change in business intelligence could remodel everything in production informatics. Along the example of Virtual Reality and a community-based granulated software ecosystem, namely the Node.js world and its interference with the System of Systems and Internet of Anything concepts, the presentation will uncover a really powerful future of Industrial software systems.**

Evolution of the Node.js world started back in 2008, when Google released the powerful V8 JavaScript engine and Ryan Dahl began using it as a general purpose virtual machine allowing millions of JS developers to create server-side or practically any computer applications for all purposes using common web technologies. Thanks to many beneficial conjunctions like the GitHub and Npm communities as well as the Node friendly PaaS (e.g., MS Azure, Google Cloud, IBM Bluemix, Heroku) today Node.js –based technologies aspire to be a common language of large-scale distributed Internet of Anything systems considering the non mission critical layers.

Our research team investigates new ideas to connect distributed Industrial system elements (sensors, actuators, control logic, intelligent machines, data logging and data mining) to each other and represents them in a Virtual World forming a general purpose information pool which allows for large-scale heterogeneous production systems.

The last part of the presentation summarizes the results and ideas of a newly developed software engine, called MAXWHERE that provides effective working environments with spatial (Virtual Reality) multimedia arrangement and Intelligent System of Systems connectivity.

The fundamental idea behind MAXWHERE is the generalization of the Document Object Model (DOM) introducing the Where Object Model (WOM) concept that covers the conventional WEB contents as well as the VR/AR building blocks in a coherent way empowered by the newest generation web APIs.

Typical applications of MAXWHERE includes industrial monitoring and facility support, context-based collaborative working environment, industrial training, and Interactive live presentations.

SHORT CV

Imre J. Rudas graduated from Bánki Donát Polytechnic, Budapest in 1971, received the Master Degree in Mathematics from the Eötvös Loránd University, Budapest, the Ph.D. in Robotics from the Hungarian Academy of Sciences in 1987, while the Doctor of Science degree from the Hungarian Academy of Sciences in 2004. He received Doctor Honoris Causa degree from the Technical University of Košice, Slovakia, from "Polytechnica" University of Timisoara, Romania, from Óbuda University, and from Slovak University of Technology in Bratislava. He was awarded by the Honorary Professor title of Wroclaw University of Technology in 2013.

He is active as a full university professor. He served as the President of Budapest Tech from 2003 till 2010. He was the founder of Óbuda University, the successor of Budapest Tec and was elected as the first President in the period 2010-2014. He served as the President of the Hungarian Rector's Conference and member of European University Association in 2008.

Now he is the Head of the Steering Committee of the University Research and Innovation Center. He has been the president of the Central European Living Lab for Intelligent Robotics since 2014.

He is a Fellow of IEEE, Senior AdCom member of Industrial Electronics Society (IES), he served IES as a Vice-President in 2000-2001. He was elected as the Vice-President for Membership and Student Activities in IEEE System, Man and Cybernetics Society for the period 2015-2016. He is the Senior Past Chair of IEEE Hungary Section.

He served IFSA (International Fuzzy System Association) as Vice-President and Treasurer for a period of 7 years; he had been the President of Hungarian Fuzzy Association for ten years. He had been the Vice-President of the Hungarian Academy of Engineers for four years.

He serves as an associate editor of some scientific journals, including IEEE Transactions on Industrial Electronics, member of editorial board of Journal of Advanced Computational Intelligence, Editor-in-Chief of Acta Polytechnica Hungarica, member of various national and international scientific committees. He is the founder of the IEEE International Conference Series on Intelligent Engineering Systems (INES), IEEE International Conference on Computational Cybernetics (ICCC), IEEE International Symposium on Computational Intelligence and Informatics (CINTI, since 2000), IEEE International Symposium on Machine Intelligence and Informatics (SAMI, since 2003), IEEE International Symposium on Intelligent Systems and Informatics (SISY, since 2003), IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI, since 2004), IEEE International Symposium on Logistics and Industrial Informatics (LINDI, since 2007). He has served as General Chair and Program Chair of numerous scientific international conferences.

His present areas of research activities are Computational Cybernetics, Robotics, Cloud Robotics, Internet of Anything, Soft Computing, Fuzzy Control and Fuzzy Sets. He has edited and/or published 22 three books, published more than 800 papers in international scientific journal, conference proceedings and book chapters, and received more than 2000 citations.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Deep learning powered automated tool for generating image based datasets

Marko Arsenovic, Srdjan Sladojevic, Andras Anderla, Darko Stefanovic, Bojan Lalic
University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia
arsenovic@uns.ac.rs, sladojevic@uns.ac.rs, andras@uns.ac.rs, darkoste@uns.ac.rs, blalic@uns.ac.rs

Abstract— **During the last decade, rapid growth of the social media usage and the Internet in general led to the drastic expansion of the publicly available data. This gives the great opportunity for research and analysis in many fields of machine learning and data science. Even though this data is open to use, often it is distributed over the various sources on the web and in order to be used in data science experiments it requires to be gathered, properly labeled and pre-processed. This is the essential step of every research and it is generally time consuming. In this paper, the automated system for gathering and preparing the dataset is proposed. The system consists of numerous features which embrace cutting edge approaches in order to reduce the time for developing the dataset. It is primarily developed for image based datasets, and provides possibility for researchers to collect the images from the different sources: social media by official APIs or online image libraries using web scraping techniques. Along with the data gathering, proposed system provides an intelligent image labeling feature by using state-of-the-art deep learning methods which use trained models for object recognition and label them in standardized formats. The system is modular and could be extended for desired objects detection by simply adding new trained models. Finally, it provides the possibility to speed up the researchers by setting their focus more on the essential parts of their studies than to generate the dataset by themselves.**

Keywords—image retrieval; deep learning; dataset tool;

I. Introduction

Quality and quantity of data usually has a great influence in research results. This makes collecting data one of the essential steps in the research process. Nowadays, there are many open resources for researchers where they can obtain the data; from many specialized websites, blogs and social media streams or even via plain queries through the search engines. This data is often provided in different formats and usually not ready for the immediate usage. Because of that, collecting data could require using various web scraping techniques and application programming interfaces (API). This makes gathering data process laborious and time consuming.

Along with collecting data, it's pre-processing and labeling is commonly required as one of the important steps in data analysis study. Applying different pre-processing methods on large datasets demands using various specialized libraries or frameworks. Also, labeling large datasets is mostly done manually by humans which is slow and sometimes prone to errors.

Due to the fact that dataset gathering and preparation could demand using various software tools and methods, the entire process could be challenging task for many researchers without decent background in information technologies. To overcome this kind of obstacles, automated system is needed.

The system proposed in this paper provides automated solution for generating large image based datasets in order to accelerate the researchers' works. Along with gathering data from many different web resources, the system supports automated image pre-processing by applying filters and transformations from specialized computer vision libraries. Due to recent advances in solving image classification and object detection tasks by applying deep learning techniques, the proposed system in this paper exploit deep convolutional neural networks (CNN) for image labeling which remarkably speed up the entire process of data preparation.

The rest of the paper is organized as follows: Section II presents related work, Section III presents the methodology, Section IV presents results and discussion, and finally, Section V holds our conclusions.

II. Related Work

Due to the rapid growth of social media platforms' usage in the last decade, large amount of publicly shared data is available which expended the possibility for various researches in many different areas.

Due to worldwide social media acceptance, most of the social media platforms provides official APIs for data collection, which gives opportunity to researchers to apply large-scale data analysis in their studies very quickly [1].

Another approach is using web scraping, automatic technique for obtaining data from web pages using specialized libraries or frameworks [2, 3]. Because of its flexibility, web scraping method is used in many data analysis studies for dataset collection phase [4, 5, 6].

Along with data retrieval from various web resources, automatic validation of data is also one of the challenging tasks. Due to the fact that many of social media platforms provide possibility to their users to annotate their images or text data by using tags, it brings chance to exploit these metadata for developing annotation-based image retrieval methods. Sun et al. in [7] proposed the framework for tag-based image retrieval which consists of five different dimensions for improving the search accuracy. Gao et al. in [8] proposed tag-based system that uses hypergraph learning approach for determination of data relevance. Lu et al. in [9] applied re-ranking images based on their visual and semantic information for their tag-based image retrieval system. Based

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

on the previous works mentioned in this section, tag-based methods proved very effective compared to the conventional methods of content-based image retrieval where some images' features are used for analyzing the visual content of the images. In the past years, deep learning methods proved highly efficient for image classification and detection tasks setting new accuracy records in many benchmark challenges [10, 11]. Wan et al. in [12] applied pre-trained deep CNN for content-based image retrieval task. The proposed method has been evaluated on five publicly available image based datasets, and showed high accuracy with more flexibility than conventional content-based techniques.

Based on the previous works mentioned in this section, authors of this paper propose novel dataset gathering system which combines both, state-of-the-art deep CNN content-based and tag-based image retrieval methods. Current systems mentioned above only focus on image retrieval, but dataset preparation process usually requires additional actions, such as pre-processing which if not handled automatically could be very laborious work. Taking that into account, along with automatic image gathering and labeling process, the automated system proposed in this paper has various ready-to-use implemented functionalities for image pre-processing, in order to speed up dataset preparation process entirely and help researchers with less software engineering knowledge to gather and prepare image-based datasets without an extra effort.

III. Materials and Methods

The complete process of designing and developing the proposed automated system for dataset preparation is explained in detail in this section.

The proposed system is a web-based and it consists of the client and server side. Server-side application is developed using microservice architecture, where each service is responsible for one segment of the proposed system including social media, labeling, image processing and web scraping. These specialized services communicate with each other using predefined APIs. Using the microservice architecture, the proposed system gains in modularity which makes maintaining and upgrading more convenient, every service could be changed or even replaced with the new one and that would not affect the rest of the system's functionality. Making integration for the new service in the system only requires registering it on the dispatcher service and implementing the communication APIs. Model of the proposed system is presented in Fig. 1.

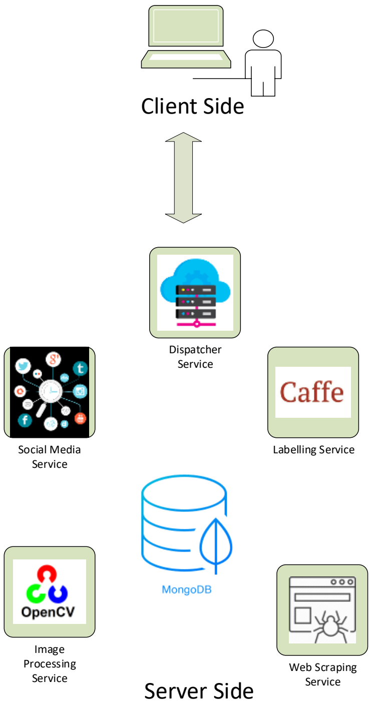
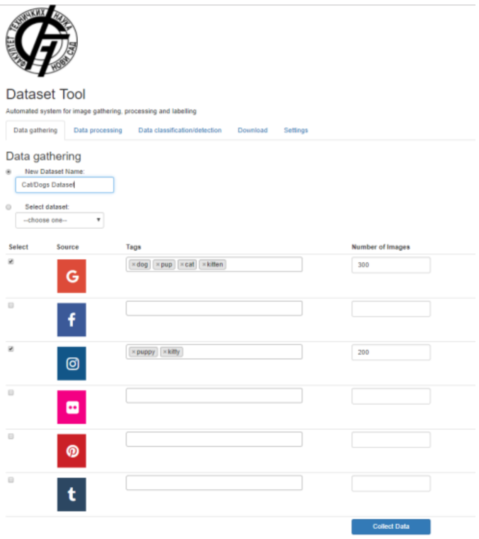


Fig. 1. Model of the system.

A. Client-side application

Client-side of the system is a single page application (SPA) which communicates with the server-side's dispatcher service via Asynchronous JavaScript and XML (AJAX) requests. All the required controls are provided to the user. One of the essential features of the proposed automated system is collecting data from various web resources. User can choose the resources from where the data is going to be fetched and also tags or queries that will be used for search, Figure 2.



Dataset Tool

Automated system for image gathering, processing and labelling

Data gathering

● New Dataset Name: Cat/Dogs Dataset

○ Select dataset: --choose one--

Select	Source	Tags	Number of Images
☑	G	×dog ×pup ×cat ×kitten	300
□	f		
☑	◎	×puppy ×kitty	200
□	••		
□	ⓟ		
□	t		

Fig. 2. Choosing data resources and tags on the client-side.

Along with choosing required data resources, user is able to apply data processing on the fetched data within the application by simply passing filters and transformations' parameters to the server-side's image processing service.

All the client's data is sent to the dispatcher service using JavaScript Object Notation (JSON) format. Based on the sent data, dispatcher service employs one or even a set of the microservices in parallel. In order to notify the clients about the server-side's processing progress, client-side application uses HTTP long polling technique to get the latest progress information. Every change of progress or even errors on the server-side are displayed inside the client-side application almost in a real time. In that way, clients are updated with the latest information.

Within the client-side application, users are able to download or delete the prepared datasets with images and metadata from the remote server. Also, there is an option for downloading larger datasets from server using file transfer protocol (FTP) with user's authentication credentials.

B. Dispatcher service

Dispatcher service is a server-side's lightweight service only responsible for handling clients' requests by calling the required services. In that way, dispatcher is not busy with background long processes and it is always open for managing new clients' requests.

This service listens to the clients' requests, handles them by raising a system threads allocated from a thread pool and based on the sent data, assigns the required services through the communication APIs. Dispatcher service is Linux hosted and written in Python, just as other services in the proposed system.

C. Web scraping and Social media services

Two methods, web scraping and social media services, are used for data acquisition. Due to the modularity and easier maintenance, each method is handled by the separate service in the system. Web scraping is handled by using specialized Python libraries and it is only developed to scrape images using client's queries on Bing and Google search engines. This method could be extended to more web resources.

Social media service exploits official APIs provided by social platforms using tag-based searches. This service is limited by the restrictions related to the number of images allowed to download per session. Social media service in this system uses APIs provided from Facebook[1], Instagram[2], Flickr[3], Pinterest[4] and Twitter[5].

Both services stop fetching the new data when the clients' desired number of images is gathered, then finishing event is triggered and user is notified on the next client's HTTP long polling progress notification request.

D. Image processing service

Image processing service implements commonly used methods for image augmentation and pre-processing. OpenCV, specialized computer vision library, is used for developing these methods [13].

Service provides methods for image filtering such as averaging, gaussian, median and bilateral filtering. Client applies these filters by changing the input parameters within the SPA.

Along with filtering methods, this service provides commonly used geometric transformations like scaling, translation, rotation, affine and perspective transformation. This service gives chance to the researchers to quickly prepare their datasets for the further usage by providing commonly used pre-processing techniques in computer vision. Service is easily extendible for the new image processing methods.

E. Labeling service

Fetching data from using only images' metadata, tags on social media streams or queries on the search engines are annotated by humans which is prone to errors and additional validation is required. In this paper, a novel approach for automated image retrieval is proposed by applying both tag-based search and content-based search.

Due to the rapid growth of successful usage of deep learning methods in many computer vision tasks, a lot of pretrained models are now provided by the academic community. When the new images are gathered by applying traditional tag-based image retrieval methods, the second stage of the dataset preparation process exploit the labeling service. It is responsible for content-based search validation by applying these state-of-the-art pretrained CNN models for image classification task. For every image, additional metadata is stored which includes the class number and the model's accuracy.

Labeling service also applies these CNN models for object detection and stores the metadata with the images in the database. For object detection task, metadata is stored in JSON file which holds the top left and bottom right points of all the bounding boxes of the classified objects in the image along with the model's accuracy.

By applying this proposed labeling service on the newly generated dataset, collected data is verified via deep learning models which improves the overall accuracy and the quality of the final dataset without the human interference.

Labeling service implemented in the proposed system uses Caffe, deep learning framework, but it can be expanded to support other widely used deep learning frameworks and libraries [14].

F. Database

All the data gathered from the proposed system is stored in MongoDB[6], open-source NoSQL document database. It is chosen for this system due to its scalability and flexibility.

Every dataset has its own collection of images and labels and it is limited to 50 GB of memory due to the hardware limitations used in this study.

[1] Facebook - https://www.developers.facebook.com/

[2] Instagram - https://www.instagram.com/developer/

[3] Flickr - https://www.flickr.com/services/api/

[4] Pinterest - https://developers.pinterest.com/

[5] Twitter - https://developer.twitter.com/en/docs

[6] MongoDB - https://www.mongodb.com/

IV. Results and Discussion

The proposed system was tested and evaluated using two separate metrics. First evaluation method included comparing time spent retrieving images of dogs and cats by human and by automated tool on five different sizes of the dataset. One of the main reasons of choosing the simple classification task is that there is no need for human expert. In most cases, human experts are vital in manual classification and data verification. Highly accurate deep learning models could fill this gap by verifying the content of the dataset without human interference. The results are presented in Table 1.

TABLE I. Comparing retrieval time

Retrieval time [min]			
Human	*Tool: tag-based*	*Tool: tag and content-based*	*Number of Images*
65	14	21	***100***
103	23	34	***200***
171	41	53	***300***
283	53	71	***400***
413	74	102	***500***

Based on the results from Table 1, even though the experiment was conducted on the simple search task and on small number of images, automated system proved to be much faster in image retrieval than humans, which is important fact taking into account that usually valuable datasets for researches hold thousands of images. Significant amount of time could be saved by applying the automated tools for dataset gathering such as the one proposed in this paper.

Besides the speed of image retrieval, accuracy of the content is also essential for this type of systems. Due to that, second metric included comparison of accuracy of gathered data retrieved by the automated tool, using only plain tag-based search, which does not apply the labeling service and combining both deep learning content-based and plain tag-based search using pre-trained AlexNet [10] network for classification task, results are displayed in Table 2.

TABLE II. Comparing accuracy of search methods

Accuracy [%]		
tag-based	*tag and content-based*	*Number of Images*
86	95	***100***
81	94	***200***
84	96	***300***
88	95	***400***
84	94	***500***

Based on the results of the second experiment, method of combining tag-based and content-based retrieval techniques, introduced in this paper, showed higher accuracy than method which uses only plain tag-based approach.

Requiring pre-trained models for certain tasks could be the limitation of this approach. In fact, it can be partly overcome in some situations with the application of general pre-trained CNN models. For example, if the user requires collecting the dog breeds dataset, tag-based search can be improved by using binary classifier to validate if there is even a dog on the fetched image. In that way, using deep learning driven content-based search method could be used as a supporting system for wrongly annotated images from the web and save great amount of time in data preparation process.

V. Conclusions

In the last decade, there was a large growth in popularity of social media platforms which resulted in having big amount of data publicly available. Due to this rapid growth of social media and information technologies in general, web shared data becomes core resource for many research studies. Because of different formats, structures and limitations of APIs, gathering and preparation of data could be often time consuming and sometimes a great obstacle for many researchers without information technology background.

In this paper, new automated image-based dataset generating tool is proposed. It consists of several services which are responsible to speed up the researchers in gathering and preparing their datasets. Essential feature of the system is a novel approach in image retrieval, which consists of combining tag-based and deep learning driven content-based search methods. Along with that, system provides various image pre-processing methods.

The proposed system proved to be efficient and time saving with high accuracy in search tasks. These results motivate authors of this paper to improve the proposed system in further. Future work could include adding more deep learning models and expand the labeling service to support more deep learning frameworks as a goal to be more efficient in verifying queries and tags, which could improve plain tag-based searches. At present, a lot of specialized deep learning pre-trained models are available and could be applied through the systems such as one proposed in this paper to create new big datasets for future researches and studies.

References

[1] Lomborg, Stine, and Anja Bechmann. "Using APIs for data collection on social media." The Information Society 30.4 (2014): 256-265.

[2] Glez-Peña, Daniel, et al. "Web scraping technologies in an API world." Briefings in bioinformatics 15.5 (2013): 788-797.

[3] Herrouz, Abdelhakim, Chabane Khentout, and Mahieddine Djoudi. "Overview of web content mining tools." arXiv preprint arXiv:1307.1024 (2013).

[4] Chakravorty, Subhayu, et al. "Data mining techniques for analyzing murder related structured and unstructured data." American Journal of Advanced Computing 2.2 (2015).

[5] Rao, M. Kameswara, et al. "Commodity Price Data Analysis Using Web Scraping." International Journal of Advances in Applied Sciences 4.4 (2015): 146-150.

[6] Novkovic, Milana, et al. "Data science applied to extract insights from data-weather data influence on traffic accidents."

[7] Sun, Aixin, et al. "Tag-based social image retrieval: An empirical evaluation." Journal of the Association for Information Science and Technology 62.12 (2011): 2364-2381.

[8] Gao, Yue, et al. "Visual-textual joint relevance learning for tag-based social image search." IEEE Transactions on Image Processing 22.1 (2013): 363-376.

[9] Lu, Dan, Xiaoxiao Liu, and Xueming Qian. "Tag-based image search by social re-ranking." IEEE Transactions on Multimedia 18.8 (2016): 1628-1639.

[10] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[11] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International Journal of Computer Vision 115.3 (2015): 211-252.

[12] Wan, Ji, et al. "Deep learning for content-based image retrieval: A comprehensive study." Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014.

[13] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[14] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014.

Determination of the critical congestion point in urban traffic networks: a case study

Gabriel BABAN, Alexandru IOVANOVICI, Cristian COSARIU and Lucian PRODAN

[1] ***Abstract*—At present, most of the attempts to quantify urban traffic flow have completely ignored its predominant social footprint, while literature concerning the appearance and forecasting of congestion points remains scarce. This paper presents a methodology for determining the critical point leading to congestion in urban traffic, based on analyyszing the network topology and flow using an innovative mixture of complex network metrics and state of the art traffic simulation. To validate the concepts and methodology we conducted a case study over the city of Timisoara. At around 0.5 million inhabitants we believe it provides , which provides a suitable benchmark for deriving empirical, results and qualitative interpretations.**

***Index Terms*—inteligent transportation systems, smart city, complex networks analysis, social networks**

I. Introduction and background

THE urban road networks are growing ever increasingly busy and traffic-jams have become a day-to-day practice for most commuters, while coping with the increasing and dynamic changing of traffic conditions in large cities remains largely an unsolved problem. In Copenhagen alone, traffic values have increased with over 50% since the 1980s; furthermore, in 2012 people were spending over 100,000 hours in queues, corresponding to an economic loss of more than 750 million Euros [1]. It is clear these issues require addressing quickly if the road network is to avoid collapse in the near future. Improving the efficiency over road networks can be done by re-designing the network. However, construction of new roads or expanding existing ones is hampered by restrictive European legislation, that enforces new rules concerning air and noise pollution [1]. A second, preferred approach is to optimize traffic flow.

However, optimizing traffic flow is widely considered a difficult task, acknowledged by the advent of IEEE's Transactions on Intelligent Transportation Systems (ITS). Research results transformed modern infrastructures to rely on Automated Traffic Control Signaling (ATCS) systems in order to implement DTM (Dynamic Traffic Management), which provide high configurability and precise coordination through the use of semaphores, such as SCATS, SCOOT, UTOPIA, or BALANCE to name a few [2]. All systems feature estimations of queue values and are proactive/reactive with respect to traffic changes; however, they all require a central entity that continuously monitors and adjusts traffic parameters based on patterns. Such a central entity requires qualified personnel that would make decisions should the ATCS be overwhelmed by traffic conditions and also constitutes a single point of failure (for instance, in case of power shortage). The communication bandwidth and computational power required by such a central entity are also considerable [3] Moreover, their reaction is based on fixed patterns, they are slow to react and they respond to dynamic changes in traffic such that only local optimizations are met (usually minimizing queues in intersections), while a global (or near-global) optimization remains unachieved.

At present, traffic flow is simulated using high-end software tools while traffic models are based on stochastic processes or cellular automata, for instance. aThey all share at least two common points: do not take into account the fact that urban traffic is a direct result of human activity and they leave quite to desire in terms of performance. All attempts to optimize traffic flow are completely ignoring the fact that traffic has a predominant social footprint [4] and would therefore potentially benefit from using tools from complex network analysis in order to better understand its dynamics and predict its patterns (and thus introduce intelligence). We believe further traffic optimizations are possible through a better understanding of social phenomena that heavily influences traffic. While both social and road networks are described as oriented graphs, they both share a hierarchical organization. Just like people, streets are not all equal in terms of traffic capacity and importance.

The literature remains sparse regarding the conditions that lead to congested intersections and even for a precise definition of what congestion means. We will therefore approach this phenomenon and try to give it a human dimension in an effort for better prevention. In order to better argue our novel perspective, we require topologies and related data taken from the real world, so we naturally chose our own city for this purpose. The city of Timisoara probably reunites what makes the second metropolitan area in Romania in terms of both population and urban density, placed in the western part of the country. Founded during the medieval era and lying at the crossroads of multiple cultures, the city offers infrastructural and traffic challenges worthy of a case study regarding the prerequisites and mitigation of the state of congestion.

The rest of the paper in organized as follows: in section 2 we are going to take an overview on the topic of congestion in transportation networks and the relationship with the complex network analysis and also present some of the results of our previous work, while in section 3 we are going to present the methodology of our research, the results of the applied algorithm and the results of the simulation. The paper concludes with a section dedicated to a thorough discussion of the results and the key conclusions.

[1] All the authors are with the Department of Computers, POLITEHNICA University, Timisoara, Romania. Their emails, corespondingly are: gabriel.baban@student.upt.ro,iovanalex@cs.upt.ro,cristiancosariu@cs.upt.ro, lprodan@cs.upt.ro,,

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

II. State of the art and previous work

A. Background on network congestion analysis

As more and more systems and phenomenons are modeled as networks, much interest was given to the advancement of theoretical models and experimental studies in order to explain and predict the causes of the congestion in networks and furthermore to alleviate it's consequences. Since the initial treatment based on operational analysis and mathematical modeling which were mostly reliable when applied on quasi-regular networks [5], [6] we witness a shift in paradigm towards applying numerical methods and simulation tools on large scale complex networks [7]. Much work was done in the area of computer networks [8], [9], [10]. The interest in this area is manifold: first modern networks are complex systems with a strict rules governing the correct functionality, where small disruptions can have a kind of butterfly effect on the quality of service [11]. In the same time, from a scientific point of view there are a lot of networks analysis and measurements tools which provide a rich amount of data and various conditions can be easily replicated in the laboratory [12].

Regarding urban traffic there are similar issues when dealing with congestion, but in the same time the environment is much less structured (a lot of problems are caused by drivers breaking some of the traffic rules and causing directly or indirectly jams [13]) and the main actors are social beings which act unpredictably [14]. Moreover scientific community is split when defining what does it mean for "traffic to be congested". A lot of definitions given in qualitative terms such as "when the cars are going at *really slow speed*" [15] or "when travel time is *inconceivably long* in regard to free flow" [16] while a few of them try to quantize the same empirical findings defining congestion as the situation when the arbitrary chosen QoS metrics fall bellow a specified percent in contrast to the free flow [17]. In the same time simulation and optimization software tools and methods define ad-hoc metrics which are sometime hard to replicate in a different environment and hardly can be the basis for defining or measuring change [3], [16].

Of particular interest for our research is to apply the concepts and methods from complex networks analysis for the purpose of determining the intersections most prone to congestion and deriving the measures that would prevent the emergence of congestion. However, research is scarce in relating congestion with exact topologies and network metrics, while these are also difficult to define. An important concept for describing real complex networks is centrality, which measures of the importance of the node to the networks. However, centrality is hard to define uniformly on the variety of scenarios where complex networks are applied, with many centrality metrics introduced over time.

Holmes et al. [8], [18]provide a thorough analysis of the correlation between the centrality metric and the probability of congestion. They use scale-free networks with broad betweenness centrality distribution. Zho et al. [19] focus on road traffic and approach the same problem by experimenting with simulation tools in order to find the phase transition and the trigger that causes the change of the normal free flow to congested traffic. The authors find a strong correlation between the degree of the node and the congestion (the more connected the node the higher the probability of congestion) and also between the betweenness centrality and the probability of congestion. It's worth mentioning the fact that betweenness centrality is a metric that describes the number of shortest paths traversing a specific node out of the total number of shortest paths which exist. They conclude that "random and scale-free networks are more tolerant to congestion than trees and regular networks" which is due to the more homogenous distribution of the betweenness and provide an empirical new metric which can be interpreted as: "critical phase transition value is smaller for networks of smaller ratio of degree to betweenness for the set of most easily congested nodes". Empirically, this signifies we should provide as much processing capacity as possible (short green cycle times) to the nodes with high betweenness.

B. Previous work

Any consideration upon transport should include road vehicles, which is by far the preferred means of transportation at present, accounting for about 70% of the total and showing a growing trend for the next decades [16]. These facts pose a real challenge to the road network as far as its capacity of supporting generated traffic is concerned. The current situation is in sharp contrast with the urban infrastructure's inability to expand, which only leaves the possibility of using intelligent solutions to manage traffic flow [2]. The current state-of-the-art shows a continuous interest in this subject and also points out the lack of a generic solution [4]. To the best of our knowledge, at present there is no approach of optimizing traffic flow that would use methods pertaining to the field of complex networks and take into account the influence of social aspects as an attempt to better understand, predict and manage traffic flow. Moreover, none of the existing implementations are publicly available, i.e. they cannot be investigated or modified in any way by the general public or at least by the scientific.

Our previous studies reveal the road network can be regarded as a graph of interconnected communities, with less important streets serving the inside of the community (and making for a first level of optimiation), while most important streets serve to interconnecting communities [20], [21]. Interconnected communities assemble a second graph (in which communities become nodes) where optimizations take place at a second, higher level [20].

Our research efforts in the last years were directed towards a proof-of-concept regarding a self-adaptive intelligent traffic management system that would also account for the social dimension of everyday traffic in urban areas. Its architecture represents is based on a 3-layer hierarchy

A comprehensive case study was conducted in La Almozara district in Saragossa, Spain [78 teza cristi], where traffic simulation capabilities were combined with genetic algorithms to optimize the traffic light cycles and signal times, using several predefined parameters and a static approach [22]. As there was no established approach on how to dynamically optimize an intersection, we designed an algorithm that quickly reacts to traffic dynamics based on local heuristics. We simulated real traffic situations using the Vissim software and showed a decrease in waiting times and queue lengths at local intersection level [2].

Our algorithm finds the optimal traffic balance for all directions in a single intersection, which corresponds to a fully implemented Level 1. If no additional messages from the network are involved, continuous recalculation naturally leads to an optimal equilibrium within the intersection, which translates into minimal values for queues on any direction [23]. We have conducted a case study on a typical situation in an intersection from the city center of Timisoara, at 9 o'clock AM, on a typical working day (shown in Figure 1). After running the data using the PTV's Vissim simulator against green times derived from our algorithm, the average queue length has achieved a significant decrease, as shown by Figure 2

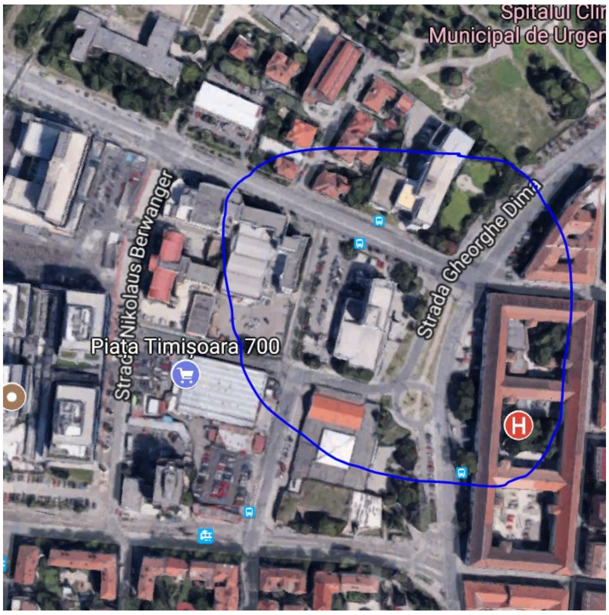


Figure 1. The case study is conducted over a section of 2 × 2km in the tech/business district of the Timisoara. The area a new developement over an old street network and not much road reconfiguration and tailoring was done to cope with the new demands.

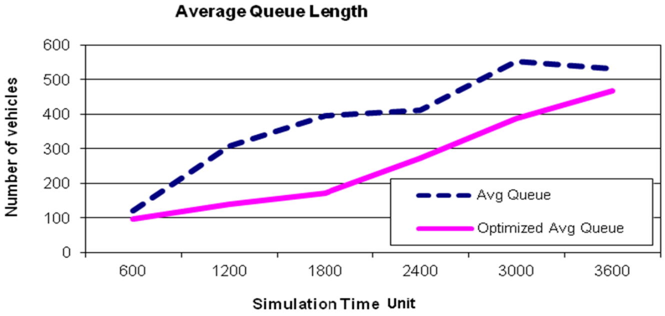


Figure 2. Appling a adaptive algorithm to the problem of traffic-light queue optimization probleme yields significant results even when dealing with large queues which, most of the time, would be considered grid-lock-ing.

III. Methodology and results

A. The dataset

We retrieved the road information data (map) from the Open Source online GIS data OpenStreetMap (OSM) [24]. The XML data are processed using a custom script we have implemented in Python and thus we create a list of all the intersections as a set of nodes and all streets inside the city as edges between nodes. This data is then assembled into a formatted gexf file, which can be directly opened in Gephi [25], an interactive platform for performing network analysis on complex systems. Gephi also features main metrics of interest: average path length, modularity and betweenness distribution. Our pipeline can handle any data set retrieved form OSM thus providing means to replicate our research

B. Complex network analysis of the road network

What we introduce in this paper is the usage of complex network analysis principles and metrics to measure and predict the properties of a road infrastructure solely based on it's topology.

We believe that the properties of a road infrastructure can be measured and even predicted by applying complex network analysis principles and metrics to its topology [20]. The data set derived in section 3.1 allows calculation of network metrics, such as network size (nodes and edges), average path length, clustering coefficient, average degree (and degree distribution), network diameter, density and modularity, and centrality, in different forms (betweenness, closeness and eigenvector) [26], [27]. The data are presented in table I and the value are consistent with the literature referenced ones for street networks. Furthermore we link traditional city neighborhoods of the city to the network using community detection and centrality algorithms.

Anywhere in our society, people commuting in regard to work or personal issues represent the bulk of our daily traffic. Moreover, a social perspective also provides innovative means of analyzing the structure of entities with a social-like structure [26]. Thus, we can detect most influential nodes (i.e. intersections) and areas with respect to traffic flow, identify patterns of communication and also study the dynamics inside the network. This strongly relates to road networks in cities [28], as it is important to determine, or even predict when possible, which areas are critical for the traffic throughput and which nodes are most influential defining possible hotspots for congestion. The results can help mitigate the congestion predict the impact of increased traffic flow. Most of the complex networks exhibit a certain community structure that has substantial importance in building an understanding regarding the dynamics of the network [26]. Therefore, our urban network analysis involves superposing and linking traditional neighborhoods of the city over the road network by using community detection and centrality algorithms [23].

We aim at finding congestion-prone intersections in a given road network, by pointing their relatively high betweenness centrality values and validating the results by using computer simulations and observation of the real traffic values. Since this is computationally and logistically difficult to perform at the scale of an entire city, we will restrict the analysis to communities (sub-graphs of the road network that map well to conventional neighborhoods). We have plotted the betweeness ditribution in figure 3 and even if it differes from a power-law as we are accustomed when dealing with complex networks, it's consistent with the poarticularities of road networks, where there are not *long-link,* which would correspond to point-to-point highways. Optimizing communities will naturally lead to a more balanced and fluent traffic city-wide. For our study we chose a community that extends over a busy business area, which is consistently affected by congestions during rush hours.

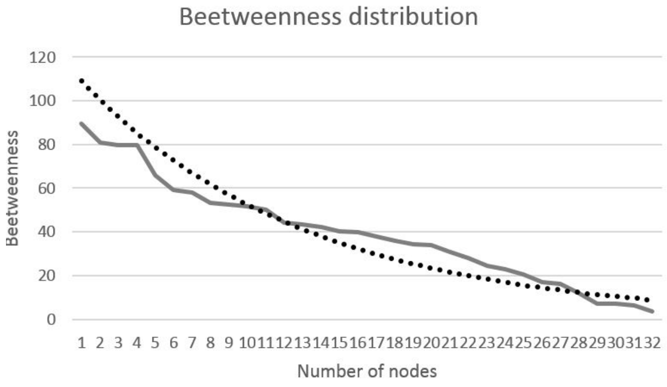


Figure 3. The distribution of the betweeness centrality metric in the case study data-set shows a typical distribution for small sized urban networks (it relates to the 2x2km area of study).

Table I
Static topological characteristics of the network presented in our study. The average degree between 1.3 and 1.4 is specific for transportation networks which are inherently flat while the average path length is correlated with the size of the road network. The high modularity number mean there is a strong, well defined, community structure.

Metric	Value
No. of nodes	42807
No. of edges	59224
No. of communities	336
Modularity	0.983
Average path length	47.094
Average degree	1.384
Population density	2.82

C. Simulation approach to congestion analysis

The second aspect of our investigation is related to the validation of the empirical findings related to betweeness centrality and congestion by means of computer simulation. Because of the large and complex system which is the urban traffic network, *in situ* actions can not be taken in order to study the effect of the control decisions. Modern investigations related to traffic design and optimization are conducted using microscopic simulators which are able to work at individual care level. We have used VISSIM form PTV, in it's scientific/research version with a study area of up to 10 × 10 km.

The first and most cumbersome step in the investigation was the set-up phase which consisted of designing the road network based on actual GIS maps and terrain images and furthermore the acquisition of raw traffic volume data. In this paper we are going to discuss the results on a smaller area of our study from typographical reasons, and more precisely we are relating the rest of the paper to figure 1, which shows a portion of the city center and business/tech district. The corresponding modeling in VISSIM is shown in figure 5 where we show the aspect of running the simulation with traffic values corresponding to a typical load for 9 AM during a weekday.

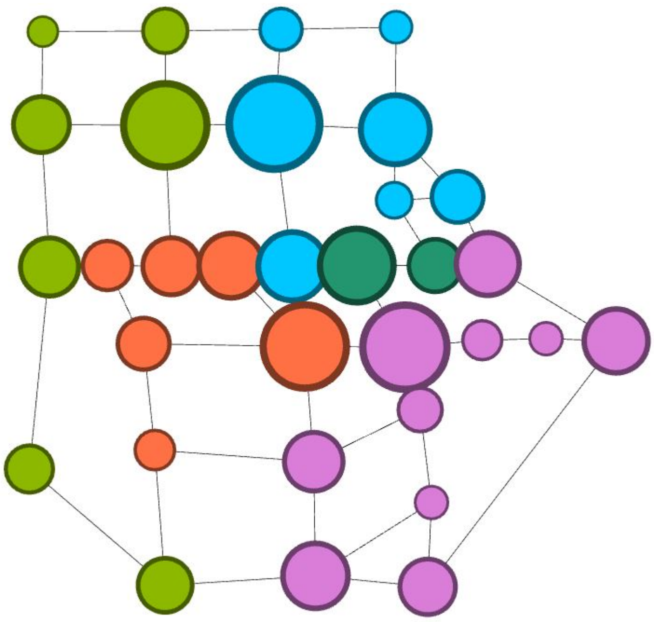


Figure 4. Network representation of the area under discussion in this paper allows us to determine specific metrics while providing in the same time spatial representation and visualization of the data. Each node corresponds to a road junction and the edges represent streets. In this figure, the nodes are scaled proportioanly to their relative betweeness centrality. We have run the community detection algorithm on the network and have mapped the identified communties (with default resolution value of 1.0) to sub-area (micro-neighborhood) of the urban networks. Each of the colors is assigned to a distinct community.

Congestion is associated with long queues and start-stop traffic so we looked at these two metrics. Simulations were run using PTV VISSIM, with queue lengths and number of stops as output data. The highest betweeness values are shown in Figure 77 , together with queue length and number of stops in the queue. Three types of traffic can be delimited from this, identified by colour:

- the green box represents free flowing with short queues and few stops before crossing the intersection. This behavior is present both at the start and the end of the simulation run. At first a small number of cars has reached the intersection so queues are short and the traffic flows as expected. Reaching the end of the simulation, the injection of cars in the network is stopped, and the queues get shorter as the remaining cars cross the intersection and exit the network;
- the transition phase represented in yellow should be the flag for the adaptive algorithm to start optimizing the green times;

- the red box is the worst state of the network. Queues start to overflow into adjacent intersections due to the limited space between intersections. A classical phenomenon known as gridlock traffic starts to develop and the only solution to end its spreading is to limit or stop the injection of cars and let the network slowly recover.

Total travel time taken to cross the network in regard to the number of cars present is plotted in figure 6a. Longer travel times are observed in two instances. At first, when the network is almost empty, dead times at the traffic lights determines a higher value for the travel time. When the network starts to reach maximum capacity and queue lengths increase, wait times are longer for a given intersection which results in longer travel times. When an equilibrium between speen, number of cars and traffic lights green times is reached in the middle part of the plott, travel time is minimised creating a so called artificial green-wave.

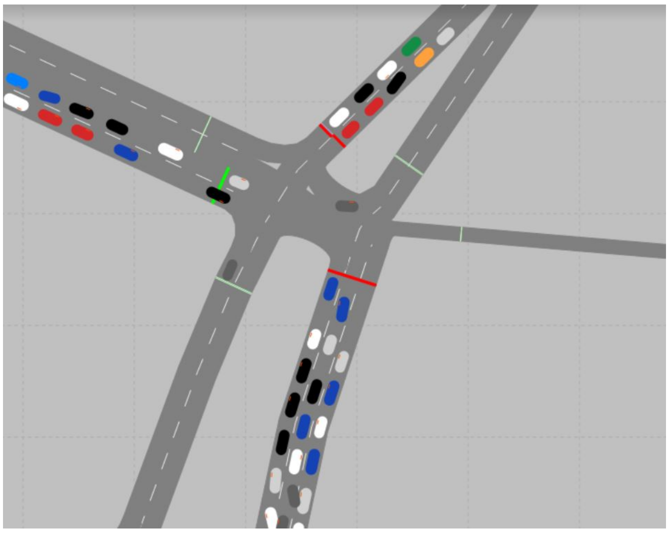


Figure 5. Microscopic simulation of the traffic flow over the intersection under study (fig. 1) provides insight on the behaviour of each of the cars, and provides means of defining cutom behaviours. The picture shows the intersection of Gh. Lazar St. in Timisoara city center.

IV. Discussions and conclusions

Our investigations were geared towards finding a better understanding of the underlying street topology and it's impact over the congestion in urban road traffic. Literature is scarce and mostly inconsistent towards finding an uniform definition of congestion and even more there are lacking formal definitions and quantifiable metrics. There is an extensive qualitative description of the phenomenon and we are "feeling" when the streets are congested but we were trying to find a numerical expression of this facts. The importance of this investigation lies at the broader scope of designing and testing self-adaptive traffic control algorithms which have to asses the possibility of reaching a grid-lock long before the point of no return.

For this we hypothesized complex network analysis could be of help mostly with it's set of centrality metrics. These try to quantize the human need of finding the most influential node in a network by algorithmic means. From the plethora of existing centrality metrics we have chosen the *betweeness centrality* because of the way it's computed: the b-centrality of a node corresponds to the number of shortest paths in the network which pass trough that node.

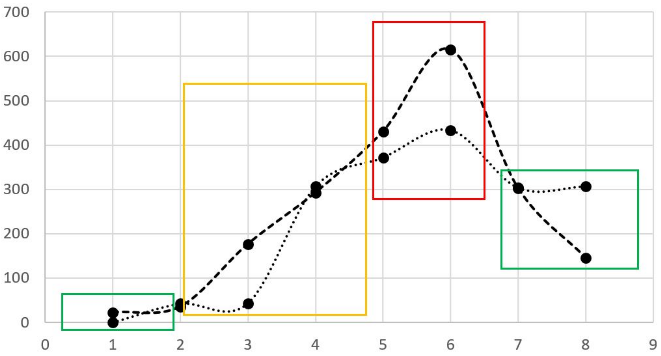
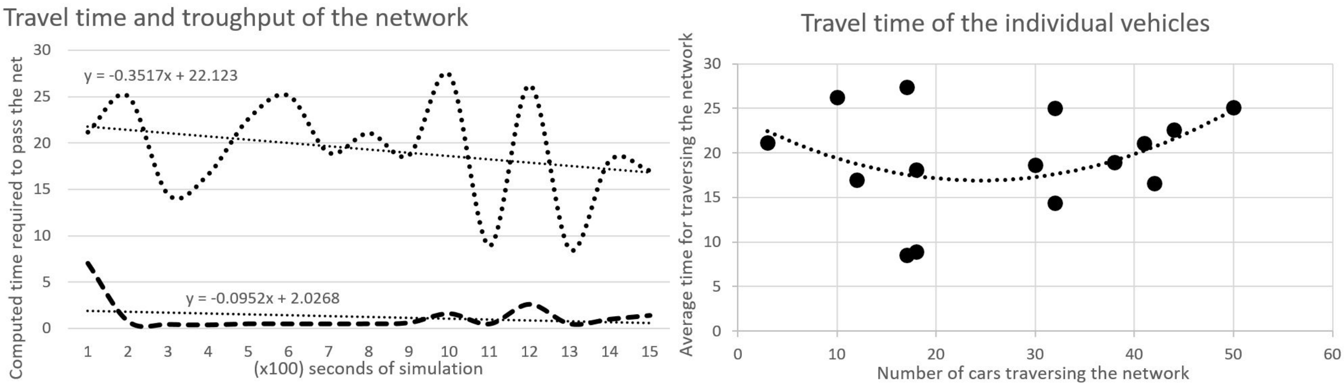


Figure 7. The dotted line represents the queue length in meters and the dashed line represents the total number of stops by all the cars in the network. These two metrics represent a good global quality indicator for the state of the network: drivers prefer not to wait at the red light and consequently the shorter the queues, the shorter the waiting time will be. This chart is derived from simulation data for the node with the highest betweenness. The practitioner can identify three types of traffic solely based on this data. The green boxes represent steady free flow traffic observable at the start of the simulation befor reaching the maximum capacity of the intersection and at the end if it when there are no new cars added to the network and the existing ones are starting the leave the network. The yellow box coresponds to the phase transition when the situation is worsening and this is the moment when adaptive alogorithms should kick into action while the red box is the worst state of the network then the quality of service degrades abruptly and in our simulation the only outcome is to cut the intake of new cars.

Several case studies were analyzed with confirmation of identification of hot spots (important intersections in terms of volume of traffic) over the city of Timisoara. Graph data analysis showed that communities identified successfully matched the neighborhoods in real life while high betweeness values point to important intersections which are susceptible to congestion.

For this case-study, we derived a workflow that uses external tools and customized scripts to parse data available online (via OpenStreetMap), so that different simulators could be used, from one specific to road traffic analysis (VISSIM) to one that is specialized in network analysis (Gephi). We tailored our STiLO algorithm to optimally determine communities and conducted a case study over a selected business community from the city of Timisoara, consistently affected by congestions. While STiLO identifies congested intersections (fig. 5) we extended our analysis using metrics from complex networks to further our understanding over the arising and disappearance of the state of congestion.

We managed to confirm the intersections prone to congestion, which matches real traffic experience. It is clear that the current state of the urban network in the area is not a good fit for achieving the state of equilibrium shown in figure 7. We would like to investigate upon the possibilities to design modifications to the network so that this state could be achieved more frequently and for extended periods of time. Adapting lanes to raise traffic capacity over preferred directions (our SIGS algorithm being a first attempt [23]) requires intelligent coordination and control that constitute the



(a)

(b)

Figure 6. Results of the simulations over the business area considered reveal an oscillatory behavior for the value of travel time with regard to the number pf cars injected in the network (subfig. a). Even if the average speed of the cars remains almost constant, a negative slope of the trend line can be observed, which is consistent with empirical observations in the field. In subfigure (b) we have plotted the total travel time of the car through the network with regard to the number of cars. To the left, the higher times are caused by the traffic lights while the rightmost part of the graph corresponds to the congested state of the network, where cars wait most of the time at queues. The graph reaches an unstable minimum, depicted in its central part, corresponding to an equilibrium between the speed of the platoon and the traffic light cycles, which creates an artificial green-wave.

subject for future work.

Acknowledgment

The authors would like to thank PTV Group who had kindly granted access to the VISSIM traffic simulator.

This work was supported by a fellowship of the Romanian National Authority for Scientific Research and Innovation, CNCS - UEFISCDI, project number BT-2016-0015, within PNCDI III

References

[1] P. C. Stubbs, W. J. Tyson, and M. Q. Dalvi, *Transport economics*. Routledge, 2017, vol. 21.

[2] A. Stevanovic, *Adaptive traffic control systems: domestic and foreign state of practice*, 2010, no. Project 20-5 (Topic 40-03).

[3] G. E. Cantarella, G. Pavone, and A. Vitetta, "Heuristics for urban road network design: lane layout and signal settings," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1682–1695, 2006.

[4] K. Williams, *Spatial planning, urban form and sustainable transport*. Routledge, 2017.

[5] C. O. PIZANO, "Mitigating network congestion: analytical models, optimization methods and their applications," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, 2010.

[6] S. H. Low, "Analytical methods for network congestion control," *Synthesis Lectures on Communication Networks*, vol. 10, no. 1, pp. 1–213, 2017.

[7] M. K. Hanif, S. M. Aamir, R. Talib, and Y. Saeed, "Analysis of network traffic congestion control over tcp protocol," *IJCSNS*, vol. 17, no. 7, p. 21, 2017.

[8] D. Ho, G. S. Park, and H. Song, "Game-theoretic scalable offloading for video streaming services over lte and wifi networks," *IEEE Transactions on Mobile Computing*, 2017.

[9] M. Welzl, *Network congestion control: managing Internet traffic*. John Wiley & Sons, 2005.

[10] H. J. Fowler and W. E. Leland, "Local area network characteristics, with implications for broadband network congestion management," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1139–1149, 1991.

[11] T. Dong, W. Hu, and X. Liao, "Dynamics of the congestion control model in underwater wireless sensor networks with time delay," *Chaos, Solitons & Fractals*, vol. 92, pp. 130–136, 2016.

[12] S. Kim, D.-Y. Kim, and J. H. Park, "Traffic management in the mobile edge cloud to improve the quality of experience of mobile video," *Computer Communications*, 2017.

[13] J. R. Olsen, R. Mitchell, D. Ogilvie, M. study team *et al.*, "Effect of a new motorway on social-spatial patterning of road traffic accidents: A retrospective longitudinal natural experimental study." *PloS one*, vol. 12, no. 9, p. e0184047, 2017.

[14] V. Truelove, J. Freeman, E. Szogi, S. Kaye, J. Davey, and K. Armstrong, "Beyond the threat of legal sanctions: What deters speeding behaviours?" *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 50, pp. 128–136, 2017.

[15] P. Pongpaibool, P. Tangamchit, and K. Noodwong, "Evaluation of road traffic congestion using fuzzy techniques," in *TENCON 2007-2007 IEEE Region 10 Conference*. IEEE, 2007, pp. 1–4.

[16] D. J. Graham, "Variable returns to agglomeration and the effect of road traffic congestion," *Journal of Urban Economics*, vol. 62, no. 1, pp. 103–120, 2007.

[17] W. Pattara-Atikom, P. Pongpaibool, and S. Thajchayapong, "Estimating road traffic congestion using vehicle velocity," in *ITS Telecommunications Proceedings, 2006 6th International Conference on*. IEEE, 2006, pp. 1001–1004.

[18] P. Holme, "Congestion and centrality in traffic flow on complex networks," *Advances in Complex Systems*, vol. 6, no. 02, pp. 163–176, 2003.

[19] L. Zhao, Y.-C. Lai, K. Park, and N. Ye, "Onset of traffic congestion in complex networks," *Physical Review E*, vol. 71, no. 2, p. 026125, 2005.

[20] A. Iovanovici, C. Cosariu, L. Prodan, and M. Vladutiu, "A hierarchical approach in deploying traffic lights using complex network analysis," in *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*. IEEE, 2014, pp. 791–796.

[21] A. Iovanovici, A. Topirceanu, C. Cosariu, M. Udrescu, L. Prodan, and M. Vladutiu, "Heuristic optimization of wireless sensor networks using social network analysis," in *Soft Computing Applications*. Springer, 2016, pp. 663–671.

[22] C. Cosariu, L. Prodan, and M. Vladutiu, "Toward traffic movement optimization using adaptive inter-traffic signaling," in *Computational Intelligence and Informatics (CINTI), 2013 IEEE 14th International Symposium on*. IEEE, 2013, pp. 539–544.

[23] C. Cosariu, A. Iovanovici, L. Prodan, M. Udrescu, and M. Vladutiu, "Bio-inspired redistribution of urban traffic flow using a social network approach," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 77–84.

[24] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[25] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks." in *ICWSM*, 2009.

[26] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[27] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[28] D. J. Watts and S. H. Strogatz, "Collective dynamics of Śsmall-worldŠnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

Manual Techniques for Evaluating Domain Usability

Michaela Bačíková, Lukáš Galko, Eva Hvizdová
Department of Computers and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Emails: {michaela.bacikova, lukas.galko}@tuke.sk,
eva.hvizdova@student.tuke.sk

***Abstract*—Domain usability is the aspect of a particular user interface that relates to its terminology, hierarchy of terms, feature descriptions or icons, language and consistency. However, domain usability is often neglected not only by software developers, but also by many researchers. To support the development of better, domain-usable user interfaces, the main contribution of this paper is the design of novel techniques for manual domain usability evaluation. We designed five qualitative evaluation techniques. To show the viability of the designed techniques, we demonstrate some of them in the domain of gospel music.**

I. Introduction

Currently, usability gets recognized more and more in the whole world. Corporations such as Google, Apple, Amazon or Facebook realized, that creating user interfaces (UIs) is not only about the looks, but the whole user experience (UX), since in this internet time as much as one click can mean instant leave of the customer to competition.

Software IT companies aim for UX and usability in general, both home and abroad. Many times they forget to take into account the work domain of their users. The issue is more evident with the domain being more specific. The time and money requirements drives IT companies to fast development and there is no time to become deeply familiar with the work domain. This fact was already identified by multiple researchers including Chilana et al. [1] and Lanthaler and Gütl [2].

To address this problem, we introduced the concept of *domain usability (DU)* and examples to illustrate our definition [3], [4]. We also proposed a conceptual design of a method for automated evaluation of DU and we performed a feasibility analysis of approaches for analysing separate DU aspects.

Despite the suggested issues, having means of at least partially and formally evaluating existing UIs for domain dictionaries would aid the quality of existing and future UIs and improve at least the fundamental issues of domain interface terminology structure.

In order to explain the design of the evaluation techniques, understanding of DU is necessary. For the space limitations, we cannot state the full definition in this paper. However, we strongly encourage the reader to familiarize with the definition [4], [3]. DU focuses on five aspects: (1) domain content, (2) consistency, (3) world language used in the interface, (4) domain specificity, (5) language barriers and errors.

A. Research Questions and Tasks

The *main goal of this paper* is to address the problem of non-existence of any formal evaluation techniques of domain usability. The main focus of this paper is on the design of *manual* evaluation techniques. We aim to propose novel methods for manual evaluation of domain usability. Such methods would aid developers and improve the situation related to domain usability of UIs.

The task of designing such methods raises the following research questions:

RQ1: How is it possible to formally or informally evaluate domain usability of existing UIs? Is it feasible (possibly with alterations) by any existing techniques?

RQ2: How to formally measure the status of UI's domain usability?

To address the above mentioned research questions and to fulfill the main goal of this paper, we state the following research tasks:

T1: Explore the potential of existing usability evaluation techniques for domain usability aspects evaluation.

T2: Experimentally design novel methods for automatized domain usability evaluation of existing UIs.

T3: Demonstrate the designed methods to evaluate their sustainability to evaluate domain usability.

In the following sections, we will describe our design of manual evaluation techniques for each of the DU aspects. After that, we describe one general technique, not specifically related to any single DU aspect.

For demonstration, and to evaluate the sustainability of the designed techniques to evaluate DU, we performed a qualitative evaluation with 3-7 subjects with the techniques described in section III. The evaluation along with other experiments is described in section IV.

II. Related Work

The following paragraphs summarize the state of the art works directly referring to the aspects of DU. Their terminology might differ from our definition. According to our knowledge, there are no metrics of DU similar to ours in the current literature.

To begin with the *domain content* aspect, there is a lot of existing literature from different authors referring to this topic. Jacob Nielsen generally refers to the topic of domain content

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

using the concept "*textual content*" of UIs. He stresses the important feature that the system should address the user's mental model [5] [6] of the domain. Accordingly, Becker [7] claims that the less complex the textual content is the more usable the application will be.

Kleshchev [8], Artemieva [9] and Gribova [10], whose works address to the importance of ontology and domain dictionary of UIs, presented a method for estimating a UI usability using its domain models. In the experiment of Billman et al. [11] usability of an old and new application was compared. An improvement in NASA user's performance was shown in the new application that had used domain-specific terminology. Badashian et al. [12] and many of the above listed authors present the importance of designing UIs that match with the real world and correspond to their domain of use.

In addition, *consistency* is another aspect that many authors refer to. The importance of this feature is presented by Badashian et al. [12]. Aspects of *world language*, *language barriers and errors* are studied by Becker [7], who deals with UIs and their translations.

Authors Isohella and Nissila [13] address the *appropriateness* of UI terminology and its evaluation by the users. According to the authors, the more appropriate the information system's terminology is, the higher is its quality. Chilana et al. [1] stress that deeper study of the target domain is needed to provide a successful and usable product in difficult domains.

Formal languages [14] [15] [16] [17] [18] (mainly domain-specific languages - DSLs) are closely related to DU. It is important to consider the domain-specific terms, properties and relations especially when designing a DSL [19]. Annotations [20] are usually also considered a DSL. Multiple authors identified the relation between the system's model [6] to UI features [5] [21] [22] [23]. DU considerably affects programmers [24] [25] [26] [27] and helps with the analysis of different formats of formal languages [28].

The number of works mentioned in this section indicates the importance of DU and a need for DU metrics to be able to formally evaluate UIs from this point of view.

III. Design of Manual Techniques for DU Evaluation

We leverage on existing techniques and base our techniques on them. Some of them are just modifications of existing techniques for the specific needs of DU, some are completely new (such as the world language evaluation technique). All mentioned techniques are suitable for evaluating DU of a domain-specific UI by usability experts. The description of each technique follows.

A. Domain Content Evaluation Technique

This technique is heavily inspired by card sorting [29]. The aim is to see how the participants interpret various terms in the application (and/or application icons). It enables to find problematic or incomprehensible terms or icons[1] that were not understood by the participant and to design better versions based on them.

Preparation: Prepare at least 10 icons from the domain application's UI, which are to be evaluated for domain content. Create one set of cards with icons and one set of cards with icon descriptions. The cards can be in electronic or paper form based on the best suitability of use.

Sample description: 3-5 domain experts in each design iteration, who may or may not be users of the tested application.

Procedure and evaluation: The technique consists of two activities:

- In the first activity, the participant reads 10 cards with icons. The participant's task is to write or tell the meaning for each of the cards according to their discretion.
- In the second activity, we add another card set containing 10 cards with meanings represented by the icons. The task is to assign each meaning to the corresponding icon.

During the procedure, it is necessary to carefully notice the moments when the participant wavers, is not sure or does not know the answer at all, while asking questions about what the participant is thinking. All these suggestions might represent a reason for changing the domain icon.

Results interpretation: Incorrect pairings are a strong indication to change the term or icon in the target UI completely, especially if it is the case with multiple testing participants. Participants hesitating during pairing is a signal to improve the term, add a tooltip or a small description, or improve the corresponding icon.

In the design of new icons, it is possible to take inspiration directly from the participants, where the best is to ask them to draw the new icon version as they imagine it would be more comprehensible and corresponding to its functionality or meaning. The technique for evaluating domain specificity (section III-D) can also be used.

B. Consistency Evaluation Technique

Consistency evaluation exploits the essence of usability inspection techniques [30]. The aim is to find consistency errors in the UI and correct them. The evaluator does not necessary need to be a DU expert.

Preparation: Target UI under evaluation with (possible) consistency errors is necessary.

Sample description: Since the inspection is performed by a domain analyst or by the evaluated application's developer team, no sample of users is needed for this technique. In some cases, a domain expert is needed to review or to judge indecisive cases.

Procedure and evaluation: An evaluator checks the whole UI for consistency errors.

The evaluation can be performed by DU experts, but also a member of the target UI developer team who is sufficient for this task. In some situations, cooperation of both might be needed, mainly in cases when it is necessary to determine whether some components perform the same operation or not.

DU experts nor developer team members might not be fully familiarized with the target domain. For some features

[1] The experiment can be performed either with icons or with domain terms, but for the sake of simplicity, the procedure and the next paragraphs will be described with only icons

very specific for the target domain, it might not be clear whether they represent the same/similar action, or not. In such indecisive cases, domain expert might help.

Results interpretation: List of consistency errors should be corrected in the target UI. If the evaluator has to decide between two possibilities for a particular feature description, then the decision should be consulted with domain experts and selected accordingly. The evaluators can also use the technique for evaluating domain specificity (section III-D) to select the most appropriate feature description.

C. World Language Evaluation Technique

In this technique we were slightly inspired by parallel design techniques [31], [32]. The aim is to create an understandable translation in the given domain. This technique can be used when creating a new UI translation, as well as to evaluate an existing UI version.

Preparation: One language version of the tested domain application in the primary or secondary language is necessary. Translation and evaluation itself can take place using a paper or electronic form, or directly in the target UI. We recommend to provide as much context as possible for the particular translations and from this point of view, we recommend to use the latter means.

Sample description: 5 - 10 subjects for each language version in both design and evaluation group. The design group needs to be composed of individuals that are familiar enough both with the primary and secondary language. The participants should be native in the primary language if possible.

Procedure and evaluation: The experiment requires two user groups: *design* and *evaluation*.

- Each member of the *design group* gets a task to create a translation into the secondary language, given they are provided with the list of UI terms in the primary language. All participants work independently on the translations to increase the number of independent translations.
- Members of the *evaluation group* get a list of all translations (created by all members of the design group). For each term they select the most suitable translation from the list. The goal is to create a new translation version by combining all translations. During this step. selection can be consulted with the evaluator.

The same procedure is used to evaluate the *fitness* of the translation vice-versa. The created translations are assigned to another two groups of users to translate into the primary language and to evaluate that the translations are understandable enough.

After the tests, a discussion is at place, where the evaluator asks the participants why they think their translation is better for the particular term. In this part, procedure similar to domain specificity evaluation technique (section III-D) is applied.

Results interpretation: This technique is more suitable for the design and prototyping phase of software development. When creating the final language version, interpretation of the results is similar to the domain specificity technique (section III-D): based on the results and discussion with the participants, an appropriate translation for the particular term is selected.

D. Domain Specificity Evaluation Technique

This technique is heavily inspired by focus group techniques [33]. The aim is to initiate a debate on domain specificity of the evaluated UI. Based on the discussion, it is possible to clarify certain domain-specific features and terminology of the target UI with the help of domain experts.

Preparation: The moderator of the discussion of a focus group prepares a list of such domain-specific features in the evaluated UI, whose description (s)he needs to clarify with domain experts, or which they feel are less understandable. Suitability of a term can be determined by using the domain content evaluation technique (section III-A).

Sample description: Similarly to standard focus groups, the group should be composed of 3 - 10 subjects. To successfully obtain the desired results for a specific domain, it is necessary for the subjects to be domain experts.

Procedure and evaluation: A term in the domain application is presented to the focus group and the group is encouraged to express their views on its suitability for the particular feature it describes. The aim is to determine if the term is too specific or too general for the particular domain. After every term, the focus group is submitted to the question whether they would suggest a more apt term. If there is any term the group did not understand, they are encouraged to discuss it. Similarly to focus groups, the moderator's task is to lead the discussion in such a way that their members were free to express their opinions and freely present their thoughts and ideas. The moderator also should direct the discussion at more general or more specific alternatives to the presented term (e.g. *"car"* and: *"automobile"* (another variant), *"vehicle"* (more general), *"electromobile"* (more specific)). This technique can also be focused on UI icons and their suitability for describing the particular feature.

Results interpretation: Based on the domain experts findings, the presented term is either replaced with a more appropriate variant suggested by the focus group, kept intact or replaced with a more general or more specific term suggested by the focus group.

E. Technique for Evaluating Language Errors and Barriers

Similarly to the consistency evaluation technique, this technique is inspired by inspection usability methods [30]. The aim is to find language errors and barriers in the UI and to correct them. The evaluator does not necessary need to be a DU expert.

Preparation: Target UI under evaluation with (possible) language errors and barriers is necessary.

Sample description: Since the inspection is performed by a DU evaluator or by the evaluated application's developer team, no sample of users is needed for this technique. In some cases, domain expert is needed to review or to judge indecisive cases.

Procedure and evaluation: An evaluator checks the whole UI for language errors and/or barriers.

The evaluation can be performed by DU experts, but also a member of the target UI's developer team is sufficient for this task.

DU experts nor developer team members might not be fully familiarized with the target domain. In case of domain-specific terms unknown to them, a domain expert is necessary to help with correction.

Results interpretation: List of errors should be corrected in the target UI. If the evaluator has to decide between two possibilities for a particular term, then the decision should be consulted with domain experts and selected accordingly. The evaluators can use the technique for evaluating domain specificity (section III-D) to select the most appropriate feature description.

F. Filling Words Technique

This technique is designed to select the most appropriate terms for an already designed UI. It uniquely enables to interconnect ergonomics with DU. The goal of this technique is to determine the understandability of the target UI based on the surrounding perceptions. It can also be used in the design phase to suggest appropriate domain terminology by using a graphical prototype of the target UI.

Preparation: It is necessary to prepare a prototype of the target UI *without domain terminology*. In problematic parts, it is appropriate to help the participant by a more general content (description, tooltip) that should be located in the given place.

Sample description: 3-5 participants are needed for this testing. If one wants to use this technique to determine the DU of the target UI, it is necessary to select participants from the set of UI users. In case of using this technique for the initial UI design, domain experts are needed, but it is better if they are also familiar with the very fundamental standards of UI design.

Procedure and evaluation: The prototype of the UI with missing domain terminology is given to the user in electronic, paper or other form. The task of each participant is to fill in the missing terminology into the UI in the language of the target interface. From the participants' proposals the most suitable terms are selected.

Results interpretation: If the participant wavers when filling any of the missing words, it might be the sign of inappropriately designed UI or incorrectly prepared application for this testing. Depending on the manifestations of the participant during testing, changes of the target UI might be needed. For further specification of the most appropriate term, the *domain specificity evaluation technique* (section III-D) can be used.

IV. Experimental Demonstration of the Designed Techniques

Four techniques designed in the previous section were demonstrated on a group of 3-7 subjects to evaluate their sustainability to evaluate DU. The numbers of subjects is variable since we were not able to get enough participants for each demonstration.

TABLE I
ICONS OF THE *Worshipper* APPLICATION

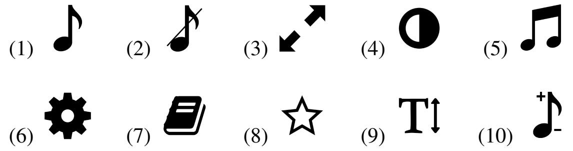


TABLE II
CARDS WITH THE DESCRIPTION OF THE ICONS FROM THE *Worshipper* APPLICATION'S UI

Songs with notes	Songs without notes	Scale to full-screen	Night mode	Song list
Settings	Song-book	Favour-ites	Change font size	Transpo sition

For the needs of the demonstrations, we chose the domain of gospel music. All selected subjects were familiar with the selected domain and their inclusion in each experiment was performed with emphasis on non-interference. In most of the tests, we used the *Worshipper*[2] Android application. Since the application is targeted at a specific domain of gospel music, and it is still in development, it is suitable for our experiments.

In cases where the technique required an erroneous UI, we modified the Worshipper's UI specifically for the needs of the particular technique. All demonstrations took place according to the described procedures in section III. In the experimental demonstrations we were also interested in *how the particular DU aspect affects the successful usage* of the UI and *user performance*.

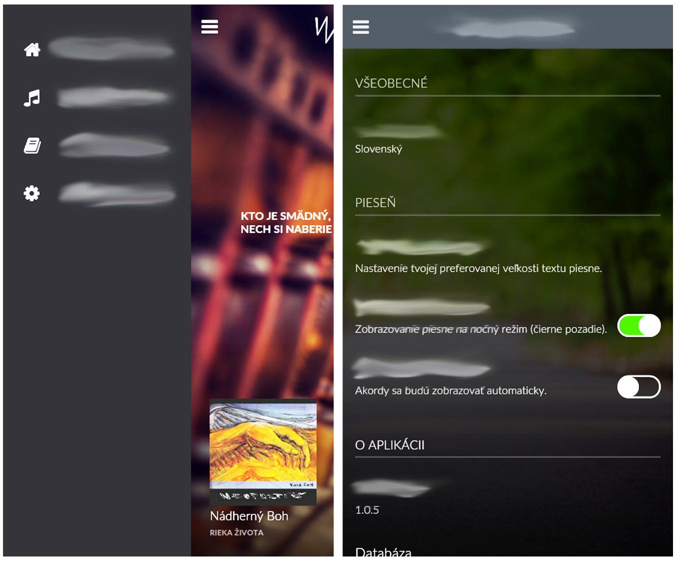
A. Domain Content Evaluation Technique

Our demonstration took place according to the described procedure. We tested 5 subjects. In the first part, the participants were given 10 cards with icons from the *Worshipper* application's UI (Tab. I).

The participants wrote the meaning of the given icons according to their discretion in Slovak language. In the second part they were given another 10 cards with icon descriptions (Tab. II).

Results: Thanks to this method it was easily possible to determine cases where the particular icon clearly specifies the UI item meaning. We also identified a few icons that

[2] https://play.google.com/store/apps/details?id=com.evey.Worshipper



VŠEOBECNÉ

Slovenský

PIESEŇ

Nastavenie tvojej preferovanej veľkosti textu piesne.

Zobrazovanie piesne na nočný režim (čierne pozadie).

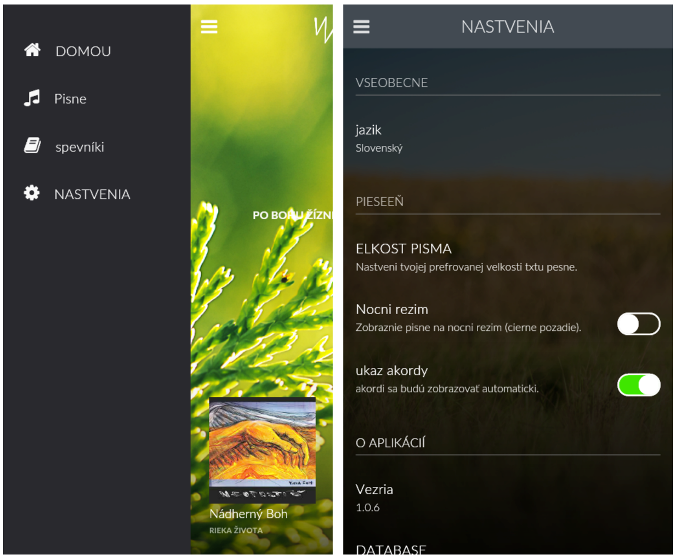
Akordy sa budú zobrazovať automaticky.

O APLIKÁCII

1.0.5

Databáza

Fig. 1. Screens of the *Worshipper* application with blank places to fill suggested terms in.



NASTVENIA

VSEOBECNE

jazik
Slovenský

PIESEEŇ

ELKOST PISMA
Nastveni tvojej prefrovanej velkosti txtu pesne.

Nocni rezim
Zobraznie pisne na nocni rezim (cierne pozadie).

ukaz akordy
akordi sa budú zobrazovať automaticki.

O APLIKÁCIÍ

Vezria
1.0.6

DATABASE

Fig. 2. Screens of the *Worshipper* application with language barriers and errors. Most common errors were: i/y replacements (*jazyk* -> *jazik*), u/v replacements at the end of a word (*DOMOV* -> *DOMOU*), doubling letters (*PIESEŇ* -> *PIESEEŇ*), switched letters (*Verzia* -> *Vezria*), missing letters (*VEĽKOSŤ* -> *ELKOST*), missing interpunction (*ukáž* -> *ukaz*).

needed a new design. An added value for the subjects was the entertaining nature of the testing. Some icons were not understandable enough for the users and in this case we asked them how they imagine the new icon design. Their drawings did not coincide very much, but each of them helped us to understand the creation of an ideal design of the particular icon.

B. World Language Evaluation Technique

In our demonstration, Slovak language was the primary language and English was the secondary one. We included 5 participants in the translation group and 7 participants in the evaluation group.

Participants from the translation group were given 4 screens of the *Worshipper* application on paper with empty places to fill English language version in. At the same time they were given a mobile phone with a working prototype, with full translation in Slovak language. Based on the Slovak terminology they were to fill English translation into the application.

From the proposals we selected the most frequent translations for each term. If the translations were ambiguous, we selected the most appropriate ones according to our discretion.

Participants from the evaluation group were given the same screens with blank places to fill in, and functional mobile application with the English translation (created by merging the results of the previous procedure). After that we reviewed the resulting translations and selected the best ones on the consultations with the subjects.

Results: Many translation suggestions were created using this method, which contributed significantly to the quality of the final translation. On the other hand, this method is very time-consuming both for the subjects and evaluators.

C. Domain Specificity Evaluation Technique

We assembled a group of 3 English-speaking musicians. Thus, naturally, the discussion was aimed at the English translation of the *Worshipper* application. We focused on terms such as "playlist", "songbook", "transposition", "chords", "notes", etc. and their appropriateness in the domain of gospel music.

Results: After the experimental procedure we assess this technique as very suitable for testing with limited time and money resources thanks to low cost of additional result processing. The solution is usually formed directly during the focus group. Preparation is also very costless, similar to the focus group method. Moreover, thanks to multiple experts directly from the target domain we were able to creatively and effectively design a suitable solution for the needs of the *Worshipper* application.

D. Filling Words Technique

The participants were given paper-printed screens of the *Worshipper* application with missing terminology. Menu icons and/or descriptions were kept in the screens for the participants to think about when filling the blank places. Some of the screens can be seen in Fig. 1.

The experimental demonstration proceeded according the described procedure in the domain of gospel music. We tested 6 participants. Most of them interpreted the blank places in the screens correctly and named them by the term that reflected the particular feature.

The most problematic place was the application version in the *Settings* screen (Fig. 1 on the right). We realized that the reason was this term was incorrectly included into the testing, because it belongs to the domain of computing

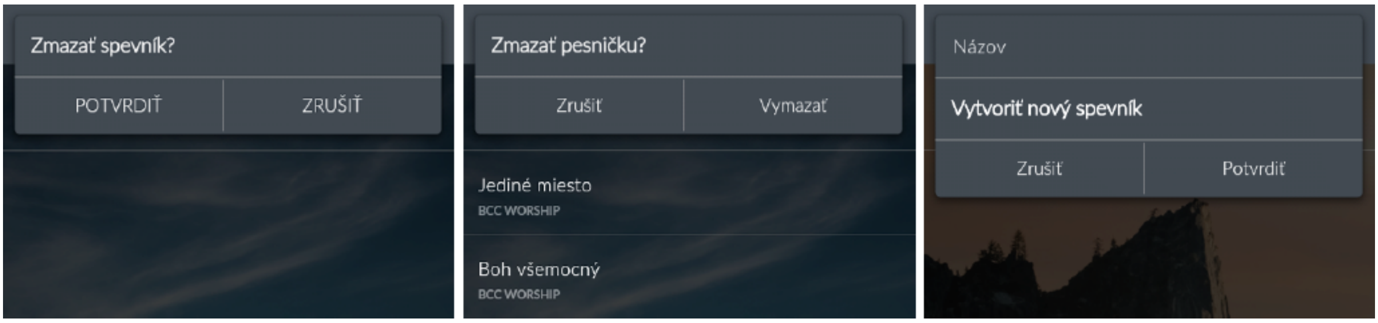


Fig. 3. Example of the *Worshipper* application with consistency errors (Zmazať spevník - Delete songbook, Zmazať pesničku - Delete song, Vytvoriť nový spevník - Create new songbook, Potvrdiť - Confirm, Zrušiť - Cancel, Vymazať - Delete.)

(or programming) and not from the domain of the target participants (gospel music). Using this method we were able ensure about the correctness of a large part of *Worshipper*'s UI terminology and at the same time we were notified about its flaws.

V. Experimental Evaluation of the Remaining DU Aspects

In this section we will describe experimental evaluation of two remaining DU aspects, demonstration of whose evaluation techniques was not described in the previous section:

- consistency and
- language barriers and errors.

Both experiments are similar: we created a functional version of the *Worshipper* mobile application with errors related to the particular DU aspect. This application was then used by the participants from the domain of gospel music in a user testing. The results were evaluated with the goal to confirm or disprove hypothesis *H*.

A. Experiment with Consistency

We created a new version of the *Worshipper* application with consistency errors in the UI. Part of this preparation can be seen in Fig. IV-D. The confirm dialogs for: (i) creating a songbook; (ii) deleting a song; and (iii) deleting a songbook; have different structure, titles and the text has different cases. The same action for deleting an object is named by two different terms (Confirm and Delete) and their positions are switched.

The scenario for the participants had the following tasks:

1) Create a songbook with a name of your choosing,
2) Add a random song into the created songbook,
3) Delete a song from any songbook,
4) Delete a songbook.

We performed a user testing, during which 5 users worked with the modified *Worshipper* mobile application and they were assigned with the prepared tasks. During the task performance, additional questions were asked with the goal of collecting information about how much the UI is comfortable to use. During the testing we recorded the subject's performance and opinions into a pre-prepared paper form. We focused mainly on the problematic parts of the application.

Results: During the testing, we noticed wavering when executing the given tasks in all cases. One of the subjects even failed to execute the task correctly. The reactions of the users were negative towards consistency errors. The results indicate, that consistency has a strong impact on DU and overall user experience which disproves the hypothesis *H* for this DU aspect.

B. Experiment with Language Errors and Barriers

We created a functional version of the *Worshipper* application with language barriers and errors (examples can be seen in Fig. IV-D). We performed user testing with 5 participants, who were working with the modified *Worshipper* application and were given a scenario with the following 5 tasks:

1) Search for a song called 'Mária.
2) Display a songbook of favourite songs.
3) Set the font size of songs display to 20.
4) Find out the database version.
5) Change the language of the application to English.

During the user testing, we asked additional questions with the goal to find more information about their satisfaction when working with the application.

Results: The main focus of the tested subjects was more on fast task accomplishment than on understanding of the text itself. In this experiment, interestingly, all users were able to work with the application despite they noticed the errors. We assume that it is due to the inherent capacity of the human brain to filter out small errors (such as typos, grammar errors, missing or reversed letters) and interpret the text regardless of them. On the other hand, in the subsequent discussion they admitted that the errors were cluttering and distracting.

From this demonstration we can conclude that language errors and barriers *do not have a strong impact on overall usability and user performance*, but on the other side, it can create a bad image of the application, its unprofessional design and negligence, which means *there is an impact on user experience*. This partially disproves hypothesis *H* related to usability and confirms it for user experience.

The user testing technique used in this experiment is suitable for detecting such errors, that would hinder the work with the application completely. In case the goal is to find the number of language errors and barriers in the domain UI, this technique is not suitable.

VI. Conclusion

The main contribution of this paper are the presented techniques of manual qualitative evaluation of DU. The techniques were designed with the goal of improving the situation considering the low DU of existing UIs. Our future research includes the design of a specific questionnaire technique similar to System Usability Scale. We also plan to define new formal DU metrics, for which we will use the results of the experiments mentioned in this paper. First, however, the weights of particular DU aspects needs to be specified.

Acknowledgment

This work was supported by the KEGA grant no. 047TUKE-4/2016: Integrating software processes into the teaching of programming and FEI-2015-23 "Pattern based domain-specific language development".

References

[1] P. K. Chilana, J. O. Wobbrock, and A. J. Ko, “Understanding usability practices in complex domains,” in *Proc. of the SIGCHI Conf. on Human Factors in Comp. Syst.*, ser. CHI ’10. NY, USA: ACM, 2010, pp. 2337–2346. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753678

[2] M. Lanthaler and C. Gütl, “Model your application domain, not your json structures,” in *Proc. 22Nd Int. Conf. on WWW ’13 Companion*. NY, USA: ACM, 2013, pp. 1415–1420. [Online]. Available: http://doi.acm.org/10.1145/2487788.2488184

[3] M. Bačíková and J. Porubän, “Ergonomic vs. domain usability of user interfaces,” in *2013 The 6th International Conference on Human System Interaction (HSI)*, June 2013, pp. 159–166.

[4] M. Bačíková and J. Porubän, *Domain Usability, User’s Perception*. Cham: Springer International Publishing, 2014, pp. 15–26. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08491-6_2

[5] M. Nosáľ and J. Porubän, “Program comprehension with four-layered mental model,” in *13th International Conference on Engineering of Modern Electric Systems, EMES 2015, art. no. 7158420*, june 2015.

[6] J. Porubän and S. Chodarev, “Model-aware language specification with java,” in *13th International Conference on Engineering of Modern Electric Systems, EMES 2015, art. no. 7158424*, june 2015.

[7] S. A. Becker, “A study of web usability for older adults seeking online health resources,” *ACM Trans. Comput.-Hum. Interact.*, vol. 11, no. 4, pp. 387–406, Dec. 2004. [Online]. Available: http://doi.acm.org/10.1145/1035575.1035578

[8] A. S. Kleshchev, “How can ontologies contribute to software development?” in *Proceedings of the First international conference on Knowledge processing and data analysis*, ser. KONT’07/KPP’07. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 121–135. [Online]. Available: http://dl.acm.org/citation.cfm?id=2022767.2022777

[9] I. L. Artemieva, “Ontology development for domains with complicated structures,” in *Proceedings of the First international conference on Knowledge processing and data analysis*, ser. KONT’07/KPP’07. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 184–202. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22140-8_12

[10] V. Gribova, “A method of estimating usability of a user interface based on its model,” *International Journal “Information Theories & Applications”*, vol. 14, no. 1, pp. 43–47, 2007. [Online]. Available: http://hdl.handle.net/10525/655

[11] D. Billman, L. Arsintescucu, M. Feary, J. Lee, A. Smith, and R. Tiwary, “Benefits of matching domain structure for planning software: the right stuff,” in *Proceedings of the 2011 annual conference on Human factors in computing systems*, ser. CHI ’11. New York, NY, USA: ACM, 2011, pp. 2521–2530. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979311

[12] A. S. Badashian, M. Mahdavi, A. Pourshirmohammadi, and M. M. nejad, “Fundamental usability guidelines for user interface design,” in *Proceedings of the 2008 International Conference on Computational Sciences and Its Applications*, ser. ICCSA ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 106–113. [Online]. Available: http://dx.doi.org/10.1109/ICCSA.2008.45

[13] S. Isohella and N. NissilÃď, “Connecting usability with terminology: Achieving usability by using appropriate terms,” in *2015 IEEE International Professional Communication Conference (IPCC)*, July 2015, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/IPCC.2015.7235849

[14] J. Kollár, I. Halupka, S. Chodarev, and E. Pietriková, “Plero: Language for grammar refactoring patterns,” in *2013 Federated Conference on Computer Science and Information Systems, FedCSIS 2013*, 2013, pp. 1503–1510.

[15] M. Tomášek, “Language for a distributed system of mobile agents,” *Acta Polytechnica Hungarica*, vol. 8, no. 2, pp. 61–79, 2011.

[16] S. Šimoňák, “Verification of communication protocols based on formal methods integration,” *Acta Polytechnica Hungarica*, vol. 9, no. 4, pp. 117–128, 2012.

[17] J. Kollár, M. Spišiak, and M. Sičák, “Abstract language of the machine mind,” *Acta Electrotechnica et Informatica*, vol. 15, no. 3, September 2015.

[18] M. Sulír and S. Šimoňák, “A terse string-embedded language for tree searching and replacing,” *Acta Electrotechnica et Informatica*, vol. 14, no. 2, June 2014.

[19] J. Juhár and L. Vokorokos, “Exploring code projections as a tool for concern management,” *Acta Electrotechnica et Informatika*, vol. 16, no. 3, April 2016.

[20] M. Sulír and J. Porubän, “Exposing runtime information through source code annotations,” *Acta Electrotechnica et Informatika*, vol. 17, no. 1, April 2017.

[21] C. Szabó, v. Korečko, and B. Sobota, “Data processing for virtual reality,” *Intelligent Systems Reference Library*, vol. 26, pp. 333–361, 2012.

[22] E. Pietriková, “Audience response systems: Benefits & utilization,” *Acta Electrotechnica et Informatica*, vol. 15, no. 4, December 2015.

[23] V. Szabóová, C. Szabó, V. Novitzká, and E. Demeterová, “Game semantics of the transaction rollback database operation,” *Acta Electrotechnica et Informatika*, vol. 15, no. 1, April 2015.

[24] W. Steingartner and V. Novitzká, “Coalgebras for modelling observable behaviour of programs,” *Journal of applied mathematics and computational mechanics*, vol. 16, no. 2, pp. 145–157, 2017.

[25] E. Pietriková and S. Chodarev, “Towards programmer knowledge profile generation,” *Acta Electrotechnica et Informatika*, vol. 16, no. 1, April 2016.

[26] J. Šťastná and M. Tomášek, “The problem of malware packing and its occurence in harmless software,” *Acta Electrotechnica et Informatika*, vol. 16, no. 3, September 2016.

[27] A. Baláž and N. Ádám, “Peer to peer system deployment,” *Acta Electrotechnica et Informatika*, vol. 16, no. 1, April 2016.

[28] Nosáľ and J. M., Porubän, “Xml to annotations mapping definition with patterns,” *Computer Science and Information Systems*, vol. 11, no. 4, pp. 1455–1477, 2014.

[29] M. Schmettow and J. Sommer, “Linking card sorting to browsing performance – are congruent municipal websites more efficient to use?” *Behav. Inf. Technol.*, vol. 35, no. 6, pp. 452–470, Jun. 2016. [Online]. Available: https://doi.org/10.1080/0144929X.2016.1157207

[30] J. Nielsen, “Usability inspection methods,” in *Conference Companion on Human Factors in Computing Systems*, ser. CHI ’94. New York, NY, USA: ACM, 1994, pp. 413–414. [Online]. Available: http://doi.acm.org/10.1145/259963.260531

[31] J. Nielsen and H. Desurvire, “Comparative design review: An exercise in parallel design,” in *Proceedings of the INTERACT ’93 and CHI ’93 Conference on Human Factors in Computing Systems*, ser. CHI ’93. New York, NY, USA: ACM, 1993, pp. 414–417. [Online]. Available: http://doi.acm.org/10.1145/169059.169327

[32] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer, “Parallel prototyping leads to better design results, more divergence, and increased self-efficacy,” *ACM Trans. Comput.-Hum. Interact.*, vol. 17, no. 4, pp. 18:1–18:24, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1879831.1879836

[33] J. Nielsen. The use and misuse of focus groups. [Online]. Available: https://www.nngroup.com/articles/focus-groups/

Complex networks analysis of international import-export trade

Renata BOAR, Alexandru IOVANOVICI, Horia CIOCARLIE

***Abstract*—International trade represent one of the main driving forces of the economy, while there are still some patterns regarding the major players in this field. Our research attempts to uncover the global import-export market by using techniques based on complex networks analysis on publicly available data. Based on our previous research and the data regarding economic cycles, we identify the links between the globally distributed companies (zaibatsu-like) and the fluctuations of the stock market. We provide a numerical proof of the correlation between the metric of betweenness from CNA and the influence of the company on the international market..**

***Index Terms*—econometrics, global financials, complex networks analysis, social networks**

I. Introduction and background

SINCE the middle of XXth century global economy has gone trough and evolutionary trasnformation process taking it's toll on both the qualitative and quatitative side. This has part of of it's root into the grand dynamics of the economy at global level. The outcome of all these processess is seen to the outside observer trough an accelerated cross-conection and linking of the national economies, which in turn are materialized as exchanges of capital, people or goods.

One of the most traditional expresions of the relationships of between two countries of represented by the economical exchange. The global economy relied on the comerce since the ancient times. The technological advancement in the area of logistics and transportation provide the means of implementing a trully global economy of which there is a great deal of interest in the area of imports and exports.

At it's core, the economical structure of the country and the political dirction represents one of the major driving forces towards the general direction of the economical decisions, but after reaching a certain treshold there is a point of no return where the influence of the external partners (other countries with their economies) plays an even greater role. Literaure suggests the fact that at global level e have to take into account not only the principles of the free exhange market bu also the interventionist policies of the state wich can hinder the eficiency of the external comerce [1].

Our investigation is more algorithmicaly oriented and tries to apply the rules and methodologies of complex networks analysis to the investiogation of the global excehange market, in order to uncover hidden patterns into the data, and provide a better and more scientific understanding of some empirical observations.

The rest of the paper is organized as follows: in section 2 we are going to take an overview of the major scienfic findings at the crossroads between economics and network analysis and also provide some key findes from our previous works into this area while in section 3 we are going to present the methodology we used for processing the data collected form Eurostat and prepare them for suitable analysis using network tools toghether with the adequate visualizations for these data. Section 4 is dedicated for a discussion of the results form an economical perspective and to draw the main conclusions.

II. State of the art and previous work

A. Network modelling of economical processess

Many scientifically important problems can be represented and empirically studied using networks. For example, biological and social patterns, the World Wide Web, metabolic networks, food webs, neural networks and pathological networks are a few examples of real world problems that can be mathematically represented and topologically studied to reveal some unexpected structural features [2]. Most of these networks possess a certain community structure that has substantial importance in building an understanding regarding the dynamics of the network. s[3]

Most of the simillar studies start with the social and economical dynamics at macro level even going as far as considering continental level processes and the influence between the continents. In the same time classical studies treat the data as time series - which they actually are - and apply modelling and alaysis tools taken from signal analysis or other branches of mathematics and engineering. With the advent of complex networks and the developement of mathematical and computational tools we are witnessing the shifting in paradigm towards this approach wich is able to better express the relationships between entitites trough links. Using the time-domain-networks we are able to witness even the dynamics in time of processess illustrated as networks.

Most modells are cause and effect and measure the correlation between these two, sometimes ingnoring the lack of causality [4].

Abundent literature exists on what is called the “new network science” [2] in various fields of interest, including economics. Barabasi presents in his seminal paper [2] a methodology which allows us to model and represent anything which can be expressed as a *relationship between some entities* - this being actually one of the underlying powers of the graph data structure - and provides us with novel means of exploring and understating this newly structured data consisting of *new metrics* and *visualizations* of the data [3]. Instead of simply analyzing a set of numbers we are now capable of visualizing the *relationship between data* and in the same time, by applying the newly proposed metrics we can obtain a better understanding of the influence of various nodes to the network as a whole [5].

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

B. Previous work

This is not our first attempt to use complex networks analysis to model and better understand the dynamics of large scale, modern economies. Our previous work was geared more towards using the currency exchange rate as an indicator of the country's economy as a whole and trying to represent the correlation beteen the variations of pairs of currency exchnge rates [6]. The result were visually spectacular and provided for the moment the confirmation of the economical dynamics which was allready know, so that it was validated by the classical economical models.

The same ideas applied to a set of data pertaining the countries of the European Union and using as an indicator - because in this case there is a common unique currency - the GDP, were explored in our second paper [7]. In this case we have correctly identified cluster of countries with strong correlation in the dynamics of the GDP (which to an extent is a global quality indicator for the economy) and some other countries which are in the second echelon, but for which there is also a tight grouping with consistent variation in the GDP trend.

III. Methodology and results

Our data sets are take from the Eurostat website (http://exporthelp.europa.eu/thdapp/index.htm). The raw data are not suitable for our investigations, thus we had to process the data and provide alternate data structures which could be further ingested into Gephi and R. Each of the 4 anexes available trough Eurostat has 7056 data points regardint the volum of import and exports of raw materials and primary goods between the EU-28 countries. Each value represent a pair beetween two countries so we have a number of 28×28 distinct values, over 9 years, from 2008 to 2016. In the rest of the paper we'll refer to the datasets regarding each EU country trough their two letter ISO denomination. There is also a distinct data set related to the bulk of materials and goods exhanged with the rest of the international market, outside EU.

Our data are furthermore represented as graph and visualised as such. We have use directed graphs and the visualizations were done using Gephi 0.8, one of the leading OpenSource network analysis tools. This allowed us to rely on the set of on-built algorithms and metrics for complex network analysis. Some of the statistical analysis was performed used the R engine.

The complex network apporach allowed the easy spotting from the visualizaton of some imbalance between the volume of import and export. One can identify the dynamics of the exchange trough, adding weights to the edges, thus providing a new dimension to the analysis.

Our current work is mostly inspired by the research carried by Mantegna [5] which involved constructed a correlation matrix based on stock trade and transactions. His methodology involved building the corresponding graph of trade interaction on which he afterward computed the minimum spanning tree which in our case is of no interest becuse from a semantic point of view it would not add more insight into data. Tuminello et al. present in [8] an extension of Mantegna's work which involved the "generation the planar maximal graph" which was able to transport even more information to the analyst.

It easy to observe that we obtain by this approach a complete graph, (Figure 1) not yielding significant spatial interpretations. In order to obtain a non-complete graph we have to build a partial graph of the initial one and we do this by imposing a cut-off threshold on the correlation index.

Using a standard literature threshold of 0.3 absolute value in order to filter the uncorrelated sets, obtaining a *proper* graph. We further quantized the range of correlation indices in five disjoint clusters, as seen in table 3. The interval ranges and color are adopted from [5] for a consistent view across all the studies.

All the renderings of the graph presented in this research and also the computation of graph related metrics were done using *Gephi 0.8.2 beta* [9], one of the leading open source tools for networks and graph analysis.

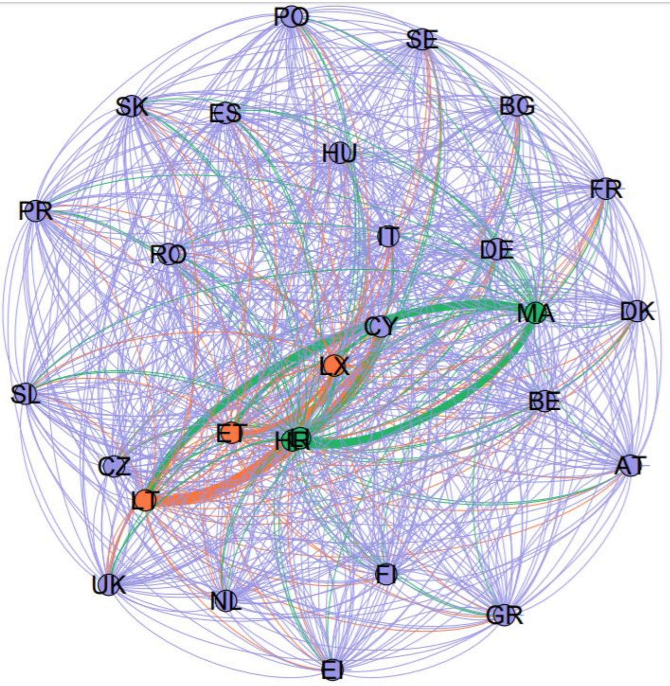


Figure 1. The import network betwwen the 28 EU countries, renedere using the ForceAtlas2 algorithm shows some tight grouping of countries with traditional good economic realtionships and a more scattered cloud of satellite-economies which have a degree of interconection but of several order os magnitude less than the big ones. This is a complete graph and the colors are assigned randomly.

Running the community detection algorithm with the parameter$resolution = 1.0$ on the filtered graph we obtain the graph shown in figure 3. There can be seen two distinct communities consisting of red and blue nodes.

Next we oriented towards the spatial visualization of the network using specific tools and algorithms. Even if for the renderings discussed up to new we have used the Fruchterman-Reingold algorithm, one can observe that it is not very useful in analyzing the structural properties of the networks being "visually messy". Of greater interest for our investigations is the Force Atlas 2 algorithm which is implemented as a standard plug-in in Gephi. Force Atlas 2's layout is based on

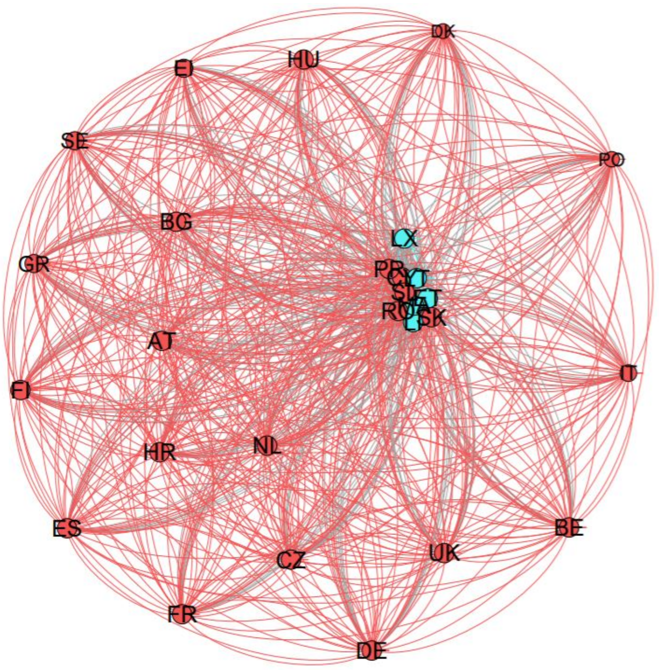


Figure 2. Regarding the export network we witness a simillar clustering provided by the ForceAtlas2 algorithm with a tight cluster of countries having the biggest slice of the volume and a constelation of less proeminent one around. This is a complete graph and the colors are assigned randomly.

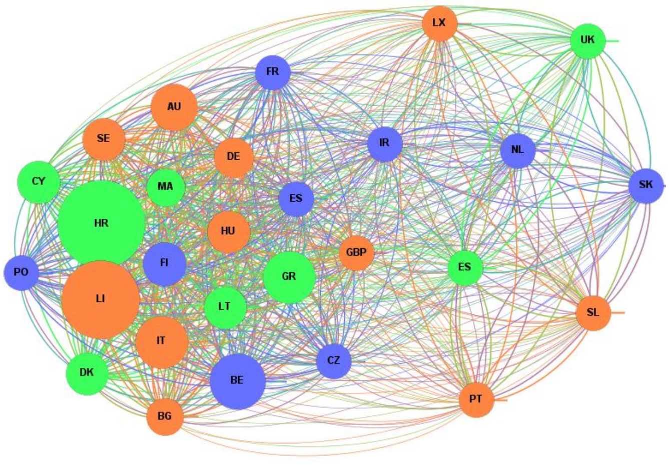


Figure 3. The graph obtained after filtering just the higher 0.7 segment of the network: jst the the upper third of the biggest countries by the volume of exchange

the analogy of gravitational forces which attract the nodes and is solving the problem as a many body system in mechanics. There are various parameters to fine tune the behavior of the algorithm, but in our case the most important one is *Edge Weight Influence* and based on the semantics of our data we swept this parameter into the range from 1 to 10, setting it for the discussion at the value of 4. The obtained rendering can be seen in figure 3 and one can perceive the spatial clustering of the two communities with only few "rogue" nodes. The economical implications of this facts are going to be discussed in the next section.

IV. Discussions and conclusions

The entirety of the exchange relationships between the countries and the major economical operators (global companies) represent what is usually known as the global market. The flow in the market is mostly explained by the flow of money, good and raw resources but also as a flow of services on a global level. A low-dimensional analysis of any of these data allows us to see a just a glimpse in the complexity: an increase or decrease in some of the metrics and maybe the correlation to the variation in some other indicator. The root cause is most of the time to well hidden in order to be able to perceive it. Also there is a need to allow a more relaxed economical policy to provide the framework for faster adaptation and counterbalance to various situations which could arise.

Regarding the European Union, the corrent EU-28, this entity carries economical and governamental policites most of the time following the mitigation and reduction of the issues and imbalances which appear in time economy at the union level.

Acknowledgment

This paper is supported by the Sectoral Operational Programme Human Resources Development POS-DRU/159/1.5/S/137516 financed from the European Social Fund and by the Romanian Government

References

[1] K.-M. Lee, J.-S. Yang, G. Kim, J. Lee, K.-I. Goh, and I.-m. Kim, "Impact of the topology of global macroeconomic network on the spreading of economic crises," *PloS one*, vol. 6, no. 3, p. e18443, 2011.

[2] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[3] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[4] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[5] P. Caraiani, "Using complex networks to characterize international business cycles," *PloS one*, vol. 8, no. 3, p. e58109, 2013.

[6] B. Renata-Graziela, A. IOVANOVICI, and H. CIOCARLIE, "Analyzing the structure or world financial markets trough complex network analysis," *CINTI*, pp. 736–740, 2014.

[7] R.-G. Boar, A. Iovanovici, and H. Ciocarlie, "Complex network interpretation of european union economic dynamics," in *Applied Machine Intelligence and Informatics (SAMI), 2017 IEEE 15th International Symposium on*. IEEE, 2017, pp. 000 201–000 206.

[8] M. Tumminello, T. Aste, T. Di Matteo, and R. Mantegna, "A tool for filtering information in complex systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 30, pp. 10 421–10 426, 2005.

[9] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks." in *ICWSM*, 2009.

[10] F. Allen and A. Babus, "Networks in finance," 2008.

[11] A. Babus, "The formation of financial networks," 2007.

[12] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, "A theory of power-law distributions in financial market fluctuations," *Nature*, vol. 423, no. 6937, pp. 267–270, 2003.

[13] C. A. Hidalgo, B. Klinger, A.-L. Barabási, and R. Hausmann, "The product space conditions the development of nations," *Science*, vol. 317, no. 5837, pp. 482–487, 2007.

[14] R. Kali and J. Reyes, "The architecture of globalization: a network approach to international economic integration," *Journal of International Business Studies*, vol. 38, no. 4, pp. 595–620, 2007.

[15] E. Kantar, B. Deviren, and M. Keskin, “Hierarchical structure of the european countries based on debts as a percentage of gdp during the 2000–2011 period,” *Physica A: Statistical Mechanics and its Applications*, 2014.

[16] D. J. Watts and S. H. Strogatz, “Collective dynamics of Śsmall-worldŠnetworks,” *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

Naive Bayes for Statlog Heart Database with Consideration of Data Specifics

Jan Bohacik
Department of Informatics
University of Zilina
Zilina, Slovakia
Jan.Bohacik@fri.uniza.sk

Michal Zabovsky
University Science Park
University of Zilina
Zilina, Slovakia
Michal.Zabovsky@fri.uniza.sk

Abstract— **Heart disease belongs to one of the main reasons for mortality nowadays and it is expected to become worse due to factors such as aging, diabetes and obesity. In addition, existing misdiagnosis of patients reporting heart related ailment worsens this situation even further. In the paper, a probability approach to recognition of heart disease is analyzed with the employment of Naive Bayes on Statlog Heart Database and with the search of data preprocessing techniques for its improvement. A discretization algorithm of numerical attributes which takes the specifics of given heart disease patients into account is presented. It is based on supervised discretization with consideration of Equal Frequency Discretization. Experiments making use of 10-fold cross-validation show improvements of accuracy which are measured with sensitivity, specificity and their sum and the results are also compared with other classification algorithms.**

Keywords—Naive Bayes; classification; heart disease

I. Introduction

Heart disease is associated with more than one disorder and any of these disorders involves the heart [11]. Coronary artery disease is the most common and it is often related to heart attack. Other disorders include difficulties of the heart to work as the pump well, heart arrhythmia, rheumatic heart disease or valves in the heart. It is a lifelong condition which stays with the patient once (s)he gets it. Sometimes people are born with it. Some of known risk factors for heart disease are smoking, high cholesterol, physical inactiveness, diabetes, high blood pressure, obesity, depression and social isolation [10]. The term "heart disease" is often used interchangeably with the term "cardiovascular disease" which also includes conditions related to narrowed or blocked blood vessels. According to World Health Organization [19], 17.5 million people die from cardiovascular disease each year, which is an estimated 31 percent of all deaths in the world. More than 75 percent of the deaths occur in low-income and middle-income countries [18]. Specifically, the poor do not have the ability to access or afford preventive services and ongoing treatments [3]. Low and middle-income countries also experience an increase in the prevalence of risk factors [2]. The signs and symptoms of heart disease depend on the specific disorder involving the heart [11]. Emergency signs are perceived to be chest discomfort, shortness of breath and fainting [16]. Other symptoms include: a) discomfort radiating to the arm, back, jaw, or throat; b) sweating; c) nausea and vomiting; e) anxiety; and f) indigestion. However, there do not have to be any symptoms at all. For example, according to [21], nearly half of all heart attacks may cause no obvious symptoms and it does not mean that these silent heart attacks are less dangerous than heart attacks with symptoms. Although people may sometimes learn about their silent heart attack accidently when the doctor is testing for something else, this is not guaranteed at all and it can be too late. Therefore, it is important to diagnose the presence of heart disease for people with high risk factors as precisely as possible and as soon as possible and give them effective treatment promptly.

Traditionally, decisions are made based on the experience of the doctor. The doctor looks for signs and symptoms and makes use of screening tests. Typical tests include measurements of the amount of cholesterol, measurements of the blood glucose level, blood pressure level, electrocardiography and exercise cardiac stress test [14]. Combining found signs, symptoms and results of the tests is not trivial and poor clinical decisions based on the experience may lead to unwanted biases and medical errors [13]. These biases and errors could be reduced by integration of clinical decision support with computer-based patient records which helps to diagnose heart disease and avoid misdiagnosis [20]. Most hospitals collect patient records in information systems nowadays but these are rarely used for automatic intelligent decision support. This decision support requires knowledge so that automatic decisions about diagnosis can be made. It might be acquired through the process of knowledge discovery in databases where patient records are utilized. Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [6]. Its computational stage is called data mining. There are several recent papers which use data mining techniques such as decision tree, Naive Bayes, nearest neighbor and neural network in relation to heart disease [1], [4], [12], [15], [22]. In [1], [4], [15] and [22], various existing data mining techniques are surveyed and compared experimentally for the purpose of heart disease diagnosis. Measures sensitivity, specificity and ROC graph are employed for comparisons in publications [15] and [22]. Paper [12] is specialized on a multilayer neural network which is trained with backpropagation and simulated on feedforward network. A probability approach with Naive Bayes and its improvement with a presented supervised discretization of numerical data are analyzed in this paper. This approach takes the specifics of given heart disease patients into consideration with the goal of increasing the diagnosis accuracy of heart disease presence.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

The following organization is used. In Section II, the used heart data and its attributes are described and analyzed in detail. The employed discretization is explained in Section III. Experiments with validation methods and achieved results are discussed in Section IV. Section V concludes the paper.

II. Statlog Heart Database

The database contains data about patients who have been subject to diagnosis of heart disease and includes their final diagnosis and physical and biochemical information about them. There are 270 patients (instances) in set $\boldsymbol{V}$ who are described by thirteen describing attributes in set $\boldsymbol{B}$ and classified into one class attribute D [8]. Attributes in the database are presented in Table I. Describing attribute $\boldsymbol{B} = \{B_1; \ldots; B_k; \ldots; B_{13}\} = \{Age;$ *Sex*; *Chest Pain Type*; *Resting Blood Pressure*; *Serum Cholesterol Level*; *Fasting Blood Sugar Over 120*; *Resting ECG Results*; *Maximal Achieved Heart Rate*; *Exercise Induced Angina*; *ST Segment Depression*; *ST Segment Slope*; *Colored Vessels*; *Thallium Heart Scan*$\}$ are defined as follows. Numerical attributes B_k can have a numerical value v for a patient $\boldsymbol{p} \in \boldsymbol{V}$ and it is marked as $B_k(\boldsymbol{p}) = v$. If possible numerical values are in set $\boldsymbol{P}$ for attribute B_k, it is denoted by $B_k = \boldsymbol{P}$. Categorical attributes B_k can have a categorical value $b_{k,l}$ for a patient $\boldsymbol{p} \in \boldsymbol{V}$ and it is denoted by $B_k(\boldsymbol{p}) = b_{k,l}$. If possible categorical values are $b_{k,1}$, …, $b_{k,l}$, …, b_{k,l_k} for attribute B_k, it is denoted by $B_k = \{b_{k,1}; \ldots; b_{k,l}; \ldots; b_{k,l_k}\}$. Attribute B_1 = Age is the age of the patient in years. *Sex* (B_2) indicates if the patient is female or male, i.e. $B_2 = \{b_{2,1}; b_{2,2}\} = \{$*female*; *male*$\}$. *Chest Pain Type* ($B_3$) is the type of chest discomfort related to a heart problem. $B_3 = \{b_{3,1}; b_{3,2}; b_{3,3}; b_{3,4}\} = \{$*typical angina*; *atypical angina*; *nonanginal*; *asymptomatic*$\}$. *Resting Blood Pressure* (B_4) is the systolic blood pressure in mmHg, i.e. how much pressure the blood of the patient is exerting against her/his artery walls when the heart beats. *Serum Cholesterol Level* (B_5) is the amount of cholesterol in the blood in mg/dL. *Fasting Blood Sugar Over 120* (B_6) indicates if the blood glucose level measured eight hours after drinking or eating is over 120 mg/dL. $B_6 = \{b_{6,1}; b_{6,2}\} = \{$*no*; *yes*$\}$. *Resting ECG Results* (B_7) indicates the electrical activity of the heart measured with the resting electrocardiogram test. $B_7 = \{b_{7,1}; b_{7,2}; b_{7,3}\} = \{$*normal*; *ST-T wave abnormality*; *left ventricular hypertrophy*$\}$. *Maximal Achieved Heart Rate* (B_8) is the maximal heart rate attained during the exercise test in bpm. *Exercise Induced Angina* (B_9) indicates if the patient gets chest discomfort with activity. $B_9 = \{b_{9,1}; b_{9,2}\} = \{$*absent*; *present*$\}$. *ST Segment Depression* (B_{10}) is related to a finding on the electrocardiogram of the patient where the ST segment is the flat, isoelectric section of the ECG between the end of the S wave and the beginning of the T wave and the depression is determined by measuring the vertical distance between the patient's trace and the isoelectric line at a location 2-3 millimeters from the QRS complex in mm. *ST Segment Slope* (B_{11}) represents the slope of ST segment during exercise. $B_{11} = \{b_{11,1}; b_{11,2}; b_{11,3}\} = \{$*upsloping*; *flat*; *downsloping*$\}$. *Colored Vessels* (B_{12}) is the number of major vessels colored by flourosopy. *Thallium Heart Scan* (B_{13}) represents the interpretation of the image of the blood flow to the heart with a special camera and a small amount of radioactive substance injected into the bloodstream. $B_{13} = \{b_{13,1}; b_{13,2}; b_{13,3}\} = \{$*normal*; *fixed defect*; *reversible defect*$\}$. Class attribute *Heart Disease* (D) classifies the patient into someone either with heart disease or without it. The attribute can have two possible values (absent/present), i.e. $D = \{d_1; d_2\} = \{$*absent*; *present*$\}$. The value of attribute D for patient $\boldsymbol{p} \in \boldsymbol{V}$ is marked as $D(\boldsymbol{p})$.

TABLE I. Attributes

Attribute	Data Type	Values	Units
Age (B_1)	Numerical	29 - 77	years
Sex (B_2)	Categorical	*female* ($b_{2,1}$)	N/A
		male ($b_{2,2}$)	
Chest Pain Type (B_3)	Categorical	*typical angina* ($b_{3,1}$)	N/A
		atypical angina ($b_{3,2}$)	
		nonanginal ($b_{3,3}$)	
		asymptomatic ($b_{3,4}$)	
Resting Blood Pressure (B_4)	Numerical	94 - 200	mmHg
Serum Cholesterol Level (B_5)	Numerical	126 - 564	mg/dL
Fasting Blood Sugar Over 120 (B_6)	Categorical	no ($b_{6,1}$)	N/A
		yes ($b_{6,2}$)	
Resting ECG Results (B_7)	Categorical	*normal* ($b_{7,1}$)	N/A
		ST-T wave abnormality ($b_{7,2}$)	
		left ventricular hypertrophy ($b_{7,3}$)	
Maximal Achieved Heart Rate (B_8)	Numerical	71 – 202	bpm
Exercise Induced Angina (B_9)	Categorical	absent ($b_{9,1}$)	N/A
		present ($b_{9,2}$)	
ST Segment Depression (B_{10})	Numerical	0 – 6.2	mm
ST Segment Slope (B_{11})	Categorical	*upsloping* ($b_{11,1}$)	N/A
		flat ($b_{11,2}$)	
		downsloping ($b_{11,3}$)	
Colored Vessels (B_{12})	Numerical	0, 1, 2, 3	count
Thallium Heart Scan (B_{13})	Categorical	*normal* ($b_{13,1}$)	N/A
		fixed defect ($b_{13,2}$)	
		reversible defect ($b_{13,3}$)	
Heart Disease (D)	Categorical	*absent* (d_1)	N/A
		present (d_2)	

Statistical analysis of the Statlog Heart Database is presented in Table II. It has the frequencies of all categorical values and the medians and modes for all particular attributes. The number of patients with absent/present heart disease is 150/120. Categorical value *male* ($b_{2,2}$) for attribute *Sex*, *asymptomatic* ($b_{3,4}$) for attribute *Chest Pain Type*, *no* ($b_{6,1}$) for *Fasting Blood Sugar Over 120* and *absent* ($b_{9,1}$) for *Exercise Induced Angina* are a lot frequenter than others. There are no missing values.

TABLE II. STATISTICAL ANALYSIS

Attribute	Value	Frequency	Median	Mode
B_1	N/A	N/A	55	54
B_2	$b_{2,1}$	87	N/A	*male*
	$b_{2,2}$	183		
B_3	$b_{3,1}$	20	N/A	*asymptomatic*
	$b_{3,2}$	42		
	$b_{3,3}$	79		
	$b_{3,4}$	129		
B_4	N/A	N/A	130	120
B_5	N/A	N/A	245	234
B_6	$b_{6,1}$	230	N/A	*no*
	$b_{6,2}$	40		
B_7	$b_{7,1}$	131	N/A	*left ventricular hypertrophy*
	$b_{7,2}$	2		
	$b_{7,3}$	137		
B_8	N/A	N/A	153.5	162
B_9	$b_{9,1}$	181	N/A	*absent*
	$b_{9,2}$	89		
B_{10}	N/A	N/A	0.8	0
B_{11}	$b_{11,1}$	130	N/A	*upsloping*
	$b_{11,2}$	122		
	$b_{11,3}$	18		
B_{12}	N/A	N/A	0	0
B_{13}	$b_{13,1}$	152	N/A	*normal*
	$b_{13,2}$	14		
	$b_{13,3}$	104		
D	d_1	150	N/A	*absent*
	d_2	120		

III. EMPLOYED ALGORITHM

It is supposed for the algorithm that previously known heart disease patients $\boldsymbol{p} \in \boldsymbol{V}$ are described by attributes $\boldsymbol{B} = \{B_1; \ldots; B_k; \ldots; B_{13}\}$ with known values and diagnosed to class attribute $D = \{d_1; d_2\}$ as they are defined in Section II. The algorithm is based on Naive Bayes which is a probabilistic approach that gives the probabilities that a patient $\boldsymbol{p}$ should be diagnosed as d_1 and d_2 as its output. The probabilities are computed as follows:

$$p(D|\boldsymbol{B}) = \frac{p(D)\prod_{B_k \in \boldsymbol{B}} p(B_k|D)}{p(\boldsymbol{B})} \tag{1}$$

where probability $p(\boldsymbol{B})$ is computed as follows:

$$p(\boldsymbol{B}) = \sum_{d_j \in D} p(d_j) \prod_{B_k \in \boldsymbol{B}} p(B_k|d_j) \cdot \tag{2}$$

Because the expression in (2) is always some constant value and only relative values of probabilities are important, probabilities for particular $d_j \in D$ are calculated in the following way:

$$p(d_j|\boldsymbol{B}) = p(d_j) \prod_{B_k \in \boldsymbol{B}} p(B_k|d_j) \cdot \tag{3}$$

The class from D with the highest probability is considered to be the class for $\boldsymbol{p}$. If the probabilities are the same, d_2 is chosen. Conditional probabilities are computed with counting occurrences of particular $b_{k,l} \in B_k$ for categorical attributes and numerical attributes are discretized at first. However, the occurrence of some value of an attribute might be zero in some cases and this would lead to the zero result of the whole expression in (3). This is not desirable as it would cause that the weight of all other attributes would not be taken into consideration. This is avoided with the Laplace estimator [17] which counts the occurrences of all possible $b_{k,l} \in B_k$ belonging to one $d_j \in D$ and adds value one to each obtained value, the total number of occurrences is increased with the cardinality of B_k , and the conditional probabilities are recomputed with obtained values.

Numerical attributes are discretized with a supervised discretization which is presented here. It makes use of entropy [7] and Equal Frequency Discretization [17] with appropriate modifications and parametrizations so that the specifics of heart disease patients and the knowledge discovered during data analysis are taken into consideration. It has the following steps for an attribute $B_k \in \boldsymbol{B}$ and D:

1. Create some initial division of the values for B_k on the basis of the heart disease patients. This should be relevant medically and the division points form some starting points of the discretization. The points are marked as $R_k = \{n_1; n_2; \ldots; n_h; \ldots; n_o\}$, i.e. there are $o - 1$ discretization subintervals.

2. Sort available data for B_k and remember particular $d_j \in D$ for each data point.

3. Compute the maximal radius for the movement of the discretization point as $r = \frac{\mathrm{argmin}_{\mathrm{all}\ h,i,h \neq i} |r_h - r_i|}{3}$ and set the smallest optimization step as $\alpha = \frac{r}{n}$ where n = 1, 2, 3, … expresses softness. Each $n_h \in R_k$ can move in both directions with multiplications of n to the maximal distance of αn (side points do not move).

4. Mark the number of possible mutual positions of points in R_k as m and do the following for all of these positions. Compute the entropy for each subinterval and find the average entropy, i. e. calculate expression $E_i = \frac{\mathbf{1}}{\boldsymbol{o}-\mathbf{1}} \sum_{\mathbf{1}}^{\boldsymbol{o}-\mathbf{1}} - \sum_{d_j \in D} p(d_j) \log_2(p(d_j))$, i = 1, 2, 3, …, m.

5. Choose the discretization which has the smallest average entropy (if there are more discretizations with the smallest entropy, choose any randomly) and transform the numerical attribute into a categorical one with this discretization.

Numerical attributes in the considered data are *Age* (B_1), *Resting Blood Pressure* (B_4), *Serum Cholesterol Level* (B_5), *Maximal Achieved Heart Rate* (B_8), *ST Segment Depression* (B_{10}) and *Colored Vessels* (B_{12}). On the basis of the domain knowledge about heart disease, division points in step 1 could lead to the following set initial discretization intervals. Attribute B_1: [0; 40), [40; 60), [60; ∞), attribute B_4: [0; 120), [120; 140), [140; 160), [160; ∞), attribute B_5: [0; 200), [200; 240), [240; ∞), attribute B_8: [0; 100), [100; 120), [120; 140), [140; ∞), attribute B_{10}: [0; 3.0); [3.0; ∞) and attribute B_{12}: [0; 1), [1; ∞). After the execution of the remaining steps of the above algorithm for all mentioned numerical attributes and $n = 2$, the following final discretization intervals are determined. Attribute B_1: [0; 45), [45; ∞), attribute B_4: [0; 107), [107; 165), [165; ∞), attribute B_5: [0; 206), [206; 216), [216; 245), [245; ∞), attribute B_8: [0; 106), [106; 114), [114; 146), [146; ∞), attribute B_{10}: [0; 2.5), [2.5; ∞) and attribute B_{12}: [0; 1), [1; ∞).

TABLE III. EXPERIMENTAL RESULTS

Method/Measure	Sensitivity	Specificity	Sum
NB-Mod	0.900	0.842	1.742
NB	0.840	0.817	1.657
MLP	0.880	0.800	1.680
DT	0.840	0.692	1.532
NN	0.773	0.717	1.490

IV. EXPERIMENTAL RESULTS

The experiments showing the performance of the algorithm presented in Section III are described here and it is compared with other well-known data mining techniques as well. The algorithm consisting of Naive Bayes, discretization and validation tools is implemented in the Java programming language. The other known data mining techniques are implemented in Waikato Environment for Knowledge Analysis [17]. The basis of the validation is *K*-fold cross-validation where *K*=10 and measures such as sensitivity, specificity and their sum are computed. *K*-fold cross-validation partitions the heart disease data into *K* subgroups with equal numbers of cases with present and absent heart disease [9]. One subgroup is used for validation while the others are used for training and this is repeated *K* times with each of the subgroups used for validation once. Sensitivity represents the ratio of patients with present heart disease who are accurately considered as the ones with heart disease. Sensitivity is computed with expression $\frac{\mathrm{tp}}{\mathrm{tp+fn}}$ where tp is the number of patients who have heart disease and who are diagnosed with heart disease and fn is the number of patients who have heart disease and who are diagnosed with no heart disease. Specificity represents the amount of patients with no heart disease who are accurately considered as patients without heart disease. Low sensitivity is associated with many heart disease patients without treatment (i.e. with life-threatening states) and low specificity is associated with useless treatment of people without heart disease (i.e. with overpriced states). The sum of sensitivity and specificity takes both life-threatening and overpriced states into consideration.

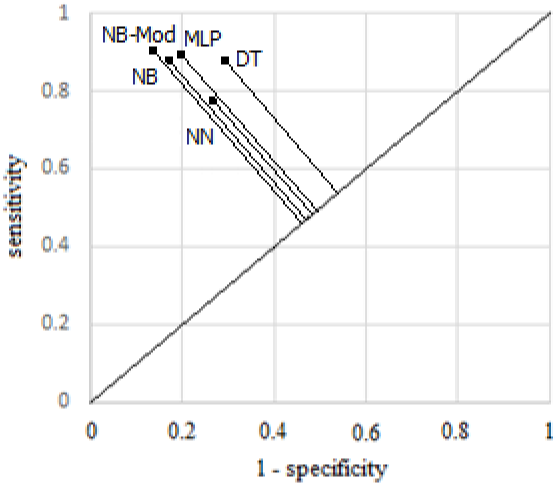


Figure 1. ROC graph.

The results of experiments are shown in Table III where the rows correspond to various algorithms and the columns are associated with computed measures. NB-Mod represents the method which is described in Section III and uses Naive Bayes with given supervised discretization which takes expert domain knowledge about heart disease patients into consideration. NB is Naive Bayes which is implemented in Waikato Environment for Knowledge Analysis as class NaiveBayes and employs well-known Fayyad-Irani's discretization [5]. MLP, DT and NN are a neural network using multilayer perception, decision tree C4.5 and a nearest neighbor classifier implemented in Waikato Environment for Knowledge Analysis as classes MultilayerPerceptron, J48 and IBk, respectively. The best results are achieved by presented NB-Mod whose sum of sensitivity and specificity equals 1.742 as a higher value of the sum is considered to be better. The ROC curve in Fig. 1 shows the results in a way which is typical for medical experts.

V. CONCLUSIONS

A decision support tool for automatic diagnosis of heart disease with a Naive Bayes classifier using a supervised discretization which takes domain expert knowledge into consideration was presented. The discretization method took initial divisions of values from experts and used them to produce the final discretization of numerical attributes. The tool is meant to support early recognition of heart disease as precisely as possible since this may be difficult due to situations with no obvious symptoms and it is important to start the treatment early without necessary costs. Statlog Heart Database was used for validation of the tool through 10-fold cross-validation with measures such as sensitivity, specificity and their sum. Sensitivity is associated with life-threatening states, specificity with costs and their sum combines both. The higher the sum is, the better the result is. With the incorporated discretization, sensitivity achieved 0.900, specificity 0.842 and their sum 1.742, which was better than well-known Naive Bayes with

Fayyad-Irani's discretization, a neural network using multilayer perception, decision tree C4.5 and a nearest neighbor classifier. Overall, the results indicated that the incorporated discretization was useful for more precise recognition of heart disease.

ACKNOWLEDGMENT

This paper was supported by the following project: “University Science Park of the University of Zilina – Phase II” (ITMS: 313011D13) of the The Operational Programme Research and Innovation funded by the European Regional Development Fund.

OPERAČNÝ PROGRAM
VÝSKUM A INOVÁCIE

Európska únia
Európsky fond regionálneho rozvoja

It was also supported by the following faculty research grant: “User-centred Approach for Decision Support Systems” (no. FVG/2/2017) of the Faculty of Management Science and Informatics, University of Zilina, Slovakia.

The authors would also like to thank Roman Trisliar for implementations of algorithms and experiments on Statlog Heart Database as a part of his studies leading to the completion of his thesis under the supervision of Jan Bohacik.

REFERENCES

[1] S. Banu, S. Swamy, “Prediction of heart disease at early stage using data mining and big data analytics: A survey,” in International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques, 2016, pp. 256 – 261.

[2] A. D. K. Bowry, J. Lewey, S. B. Dugani, N. K. Choudhry, “The burden of cardiovascular disease in low- and middle-income countries: Epidemiology and Management,” Canadian Journal of Cardiology, vol. 31, no. 9, pp. 1151-1159, 2015.

[3] F. P. Cappuccio, M. A. Miller, “Cardiovascular disease and hypertension in sub-Saharan Africa: burden, risk and interventions,” Internal and Emergency Medicine, vol. 11, pp. 299-305, 2016.

[4] S. Ekiz, P. Erdogmus, “Comparative study of heart disease classification,” in Electric Electronics, Computer Science, Biomedical Engineerings’ Meeting, 2017

[5] U. M. Fayyad, K. B. Irani, “Multi-Interval discretization of continuous-valued attributes for classification learning,” in International Joint Conference on Uncertainty in AI, 1993, pp. 1022-1027.

[6] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, “From data mining to knowledge discovery in databases,” AI Magazine, vol. 17, no. 3, pp. 37-54, 1996.

[7] R. Kohavi, J. Dougherty, M. Sahami, “Supervised and unsupervised discretization of continuous features,” in International Conference on Machine Learning, 1995, pp. 194-202.

[8] Lichman, M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA, USA: University of California, School of Information and Computer Science, 2013.

[9] McLachlan G., Do K.-A., Ambroise C.: Analyzing Microarray Gene Expression Data. San Diego, USA: Willey, 2004.

[10] National Heart Foundation of Australia, Heart Information: Coronary Heart Disease. Australia: National Heart Foundation of Australia, 2013.

[11] National Center for Chronic Disease Prevention and Health Promotion, Know the Facts About Heart Disease. : National Center for Chronic Disease Prevention and Health Promotion, 2013.

[12] E. O. Olaniyi, O. K. Oyedotun, A. Helwan, “Neural network diagnosis of heart disease,” in International Conference on Advances in Biomedical Engineering, 2015, pp. 21-24.

[13] S. Palaniappan, R. Awang, “Intelligent heart disease prediction system using data mining techniques,” in Computer Systems and Applications, Doha, Qatar, 2008.

[14] M. Pignone, A. Fowler-Brown, M. Pletcher, J. A. Tice, “Screening for asymptomatic coronary artery disease: A systematic review,” Systematic Evidence Review, no. 22, 2003.

[15] M. Sultana, A. Haider, M. S. Uddin, “Analysis of data mining techniques for heart disease prediction,” in International Conference on Electrical Engineering and Information Communication Technology, 2016.

[16] Victorian Government, Fainting and Collapse. Melbourne, Australia: Victorian Government, 2011.

[17] I. H. Witten, E. Frank, M. A. Hall, Practical Machine Learning Tools and Techniques (3rd Edition). Burlington, MA, USA: Morgan Kaufman Publishers, 2011.

[18] World Health Organization. Global Status Report on Non-Communicable Diseases 2010. Geneva: World Health Organization, 2011.

[19] World Health Organization, World Heart Federation, World Stroke Organization. Global Atlas on Cardiovascular Disease Prevention and Control. : World Health Organization, 2011.

[20] R. Wu, W. Peters, M. W. Morgan, "The next generation clinical decision support: Linking evidence to best practice," Journal Healthcare Information Management, vol. 16, no. 4, pp. 50-55, 2002.

[21] Z. M. Zhang, P. M. Rautaharju, R. J. Prineas, C. J. Rodriguez, L. Loehr, W. D. Rosamond, D. Kitzman, D. Couper, E. Z. Soliman, “Race and sex differences in the incidence and prognostic significance of silent myocardial infarction in the atherosclerosis risk in communities (ARIC) study,” vol. 133, no. 22, 2016.

[22] I. A. Zriqat, A. M. Altamimi, M. Azzeh, “A comparative study for predicting heart diseases using data mining classification methods,” International Journal of Computer Science and Information Security, vol. 14, no. 12, pp. 868-879, 2016

Nearest Neighbor Method Using Non-nested Generalized Exemplars in Breast Cancer Diagnosis

Jan Bohacik
Department of Informatics
University of Zilina
Zilina, Slovakia
Jan.Bohacik@fri.uniza.sk

Michal Zabovsky
University Science Park
University of Zilina
Zilina, Slovakia
Michal.Zabovsky@fri.uniza.sk

Abstract— **Every year there are several million people in the world who die from cancer while breast cancer is belonging to the most prevalent cancers diagnosed in women. In this paper, a nearest neighbor method which uses non-nested generalized exemplars is analyzed for diagnosis of breast cancer. The aim is to improve its accuracy so that the severity of a mammographic mass lesion is predicted more accurately from BI-RADS attributes and the age of the patient. The improvement consists in a change of distance computation between attributes with missing values and the use of several exemplars in diagnosis for a patient. Experiments on mammographic mass data make use of 10-fold cross-validation where sensitivity, specificity and overall accuracy are computed. Achieved results show increases in the sum of sensitivity and specificity as a combined measure for minimization of life-threatening situations and costs. Overall, the amount of unnecessary biopsies is decreased in the analyzed method.**

Keywords—nearest neighbor; classification; breast cancer

I. Introduction

Breast cancer is one of the most diagnosed cancers for women in Europe and its incidence is still rising mainly due to ageing population in the European Union [7]. Other risk factors include previous incidence of breast cancer, certain inherited genes, alcohol consumption, obesity and low physical activity for instance. There are almost half a million new cases diagnosed annually [2] and associated direct medical costs are estimated fifteen billion euros a year [8]. Incidence is higher in Western and Northern Europe in comparison to Eastern and Southern Europe but it is increasing more quickly in Eastern and Southern Europe due to adoption of so-called western lifestyle [6], [14]. Many people do not realize that in fact there are several types of breast cancer and so it is a group of diseases with some differences and particularly adjusted treatments. In general, growing cells in the breast out of control is considered to be the start of breast cancer [1]. Usually, these cells form a tumor or mass of abnormal tissue visible on an x-ray or touchable as a lump. If the cells invade surrounding tissues or spread to the body, this tumor is malignant. Overall, most breast lumps are not cancer and they are benign. These are just abnormal growths which do not spread to the body and so they are not considered life threatening. However, some benign breast lumps might increase the possibility of getting breast cancer. They might also become malignant later. Moreover, breast cancers do not normally show symptoms until they are at advanced stages. Therefore, it is important to detect breast cancer masses as early as possible and distinguish between benign and malignant masses. About seventy percent of women who are diagnosed with early breast cancer will eventually not develop any advanced stages according to statistical data [10].

There are several techniques developed for screening and diagnosis of breast cancer [4], [12]. A strongly recommended way how to find if there is any tumor is the examination of breasts by women themselves on a regular basis. Especially women over 40 should receive regular clinical breast exams. However, mammography has been a pillar for decades where a mammogram is actually an x-ray of the breast. There a two types of mammograms: a) screening mammograms and b) diagnostic mammograms. Screening mammograms can show breast cancer in women with no symptoms. Diagnostic mammograms are used when there are suspicious signs in a screening mammogram or after some suspicious other signs such as a lump, breast pain, nipple discharge, changes in the size of the breast etc. An important advantage of mammography is that it is a non-invasive method. But on the other hand, it lacks diagnostic accuracy. Specifically false positive situations should be minimized as they lead to unnecessary biopsies for benign situations. Biopsy gives the highest diagnostic accuracy but it is an invasive method with higher costs. Innovated biopsy methods such as core needle biopsy try to minimize its negatives. Magnetic resonance imaging is also used and its accuracy for malignant situations is quite high. However, it has high costs and its accuracy for benign situations is not that high. It is often used for high risk patients and as an addition to mammography. This is also the case for ultrasound. Mammograms are usually assessed by radiologists on the basis of so-called BI-RADS, i.e. the breast image reporting and database system [11]. These assessments support the decisions of physicians. An improvement of Nearest Neighbor Method Using Non-nested Generalized which automatically suggests if the mass for a particular patient in mammography is benign or malignant is employed in this paper. It is applied to Mammographic Mass Data. The distance computation between attributes is adjusted so that attributes with missing values are included. Several exemplars are considered when diagnosis is being made in comparison to the original algorithm as well.

The following organization of the paper is used. In Section II, the data extracted from the mammograms is described and analyzed. The adjusted algorithm utilized for suggestions about the mass is explained and presented in

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Section III. Validation methods and the achieved performance are discussed in Section IV together with exact experimental results. Finally, Section V contains the conclusion.

TABLE I. ATTRIBUTES

Attribute	Data Type	Values	Units
Age (B_1)	Numerical	18 - 96	years
Mass Density (B_2)	Categorical	*high* ($b_{2,1}$)	N/A
		iso ($b_{2,2}$)	
		low ($b_{2,3}$)	
		fat-containing ($b_{2,4}$)	
Mass Margin (B_3)	Categorical	*circumscribed* ($b_{3,1}$)	N/A
		microlobulated ($b_{3,2}$)	
		obscured ($b_{3,3}$)	
		ill-defined ($b_{3,4}$)	
		spiculated ($b_{3,5}$)	
Mass Shape (B_4)	Categorical	*round* ($b_{4,1}$)	N/A
		oval ($b_{4,2}$)	
		lobular ($b_{4,3}$)	
		irregular ($b_{4,4}$)	
Severity (D)	Categorical	*benign* (d_1)	N/A
		malignant (d_2)	

II. MAMMOGRAPHIC MASS DATA

The employed dataset can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the age of the patient [5]. It consists of 961 mammograms (instances in $\boldsymbol{V}$) gathered at the Institute of Radiology, University Erlangen-Nuremberg, Germany where there are 516 benign masses and 445 malignant masses. Each mammogram corresponds to a particular patient. Attributes in the dataset are presented in Table I. Describing attributes $\boldsymbol{B}$ = $\{B_1; B_2; B_3; B_4\}$ = {*Age*; *Mass Density*; *Mass Margin*; *Mass Shape*} collected for each mammogram are defined as follows. Attribute B_1 = *Age* is the age of the patient whose mammogram is studied in years. The particular numerical value in years for some mammogram $\boldsymbol{e}$ can be obtained as $B_1(\boldsymbol{e})$. Attribute B_2 = *Mass Density* is the density of the mass found in a particular mammogram and it is defined in terms of the relative amounts of present fatty elements. It is often denser in the middle than towards the edges. High density tends to be typical for malignant cancer cells. The highest possible density is *high*, the second highest possible density is *iso*, the third highest possible density is *low* and finally the lowest possible density is *fat-containing*. It is denoted as B_2 = $\{b_{2,1}; b_{2,2}; b_{2,3}; b_{2,4}\}$ = {*high*; *iso*; *low*; *fat-containing*}. The particular possible value for some mammogram $\boldsymbol{e}$ can be obtained as $B_2(\boldsymbol{e})$. Attribute B_3 = *Mass Margin* represents the margin of the breast mass found in a particular mammogram and it tends to be suspicious if it is ill-defined or spiculated. Blurry margins tend to indicate breast cancer cells which are infiltrating into the surrounding tissue. Possible categorical values are *circumscribed* ($b_{3,1}$), *microlobulated* ($b_{3,2}$), *obscured* ($b_{3,3}$), *ill-defined* ($b_{3,4}$) and *spiculated* ($b_{3,5}$). It is denoted as B_3 = $\{b_{3,1}; b_{3,2}; b_{3,3}; b_{3,4}; b_{3,5}\}$ = {*circumscribed*; *microlobulated*; *obscured*; *ill-defined*; *spiculated*}. The particular possible value for some mammogram $\boldsymbol{e}$ can be obtained as $B_3(\boldsymbol{e})$. Attribute B_4 = *Mass Shape* is the shape of the mass found in a particular mammogram. It usually has convex outside borders but it can be any in general. Possible categorical values are *round* ($b_{4,1}$), *oval* ($b_{4,2}$), *lobular* ($b_{4,3}$) and *irregular* ($b_{4,4}$). This is denoted as B_4 = $\{b_{4,1}; b_{4,2}; b_{4,3}; b_{4,4}\}$ = {*round*; *oval*; *lobular*; *irregular*}. The particular possible value for some mammogram $\boldsymbol{e}$ can be obtained as $B_4(\boldsymbol{e})$. Categorical class attribute D (*Severity*) classifies the mammograms into *benign* and *malignant*, i.e. D = $\{d_1; d_2\}$ = {*benign*; *malignat*}. The value for $\boldsymbol{e}$ is $D(\boldsymbol{e})$.

TABLE II. STATISTICAL ANALYSIS

Attribute	Value	Frequency	Median	Mode
B_1	N/A	N/A	57	59
B_2	$b_{2,1}$	16	N/A	*low*
	$b_{2,2}$	59		
	$b_{2,3}$	798		
	$b_{2,4}$	12		
B_3	$b_{3,1}$	357	N/A	*circumscribed*
	$b_{3,2}$	24		
	$b_{3,3}$	116		
	$b_{3,4}$	280		
	$b_{3,5}$	136		
B_4	$b_{4,1}$	224	N/A	*irregular*
	$b_{4,2}$	211		
	$b_{4,3}$	95		
	$b_{4,4}$	400		
D	d_1	516	N/A	*benign*
	d_2	445		

Statistical analysis of the dataset is summarized in Table II where the frequency of categorical values, median and mode are computed. The amount of benign masses and the amount of malignant masses are similar. Some categorical values are a lot frequenter than others. For example, *low* ($b_{2,3}$) for *Mass Density* (B_2) is very frequent. In addition to the information shown in Table II, there are missing values in the data. This can happen when these values are unknown in the case of age or when these values are not extracted for some technical reason. Attribute $B_1/B_2/B_3/B_4$ has 5/76/48/31 missing values for all instances, respectively. When the missing values are combined for all attributes in $\boldsymbol{B}$, there are 131 instances with at least one missing value and at most two missing values in total.

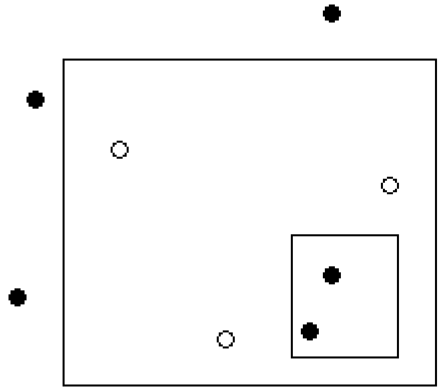


Figure 1. Nested exemplars.

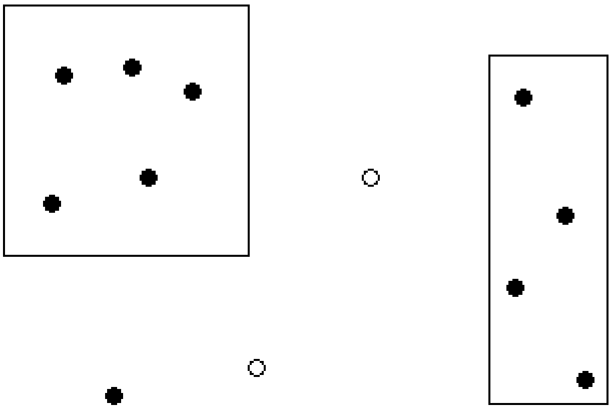


Figure 2. No-nested exemplars.

III. Employed Algorithm

Nearest neighbor method using non-nested generalized exemplars is a method that combines instance-based learning and rule induction so that the advantages of both are maintained [3]. Traditionally instance-based learning is lazy and so most computations are done during predictions. Some advantages are that it can incorporate new instances over time and it can often start working well immediately even with small amount of input data. However the amount of stored instances can become too large. Rule induction is usually eager, i.e. there are a lot of computations during learning when the data is generalized into possibly a small set of rules. Some advantage is that predictions with rules are computationally less expensive. Some other advantage is that the set of rules is much smaller than all input data. The instances or rules stored for the purposes of predictions are called exemplars. If the stored instances are generalized into rules, these rules are also called generalized exemplars (if they do not consist of just one instance). In a geometric sense, instances can be considered to be points in an n-dimensional problem space where n is the number of attributes in $\boldsymbol{B}$. A generalized exemplar is an n-dimensional region covering a finite area of the problem space and includes several points representing instances. Geometrically, this region is called n-dimensional rectangle. In the employed algorithm, it is not allowed to create generalized exemplars with overlaps and hence non-nested is used in the name of the method (see Fig. 1 and Fig. 2). During learning (i.e. rule induction), some further generalization is performed for any new considered instance by joining it to the nearest rectangle with the same severity $d \in \boldsymbol{D}$. Each prospective new generalization is tested so that no overlaps occur in rectangles. If any overlaps are found later in the learning, rules are changed so that these overlaps are removed. The particular steps of the algorithm can be described as follows:

1. If $\boldsymbol{V} = \emptyset$, end. Read mammogram $\boldsymbol{e} \in \boldsymbol{V}$ and store it, set $\boldsymbol{V} = \boldsymbol{V}\backslash\{\boldsymbol{e}\}$, adjust ranges of attributes in $\boldsymbol{B}$. Go to Step 2.
2. For each stored rectangle, compute the distance from mammogram $\boldsymbol{e}$ to the rectangle. If the computed distance is less than minimal found distance for $d \in \boldsymbol{D}$ of this rectangle, set minimal found distance of this d to the computed distance and set the count of this d to 1. Do the same with stored un-generalized exemplars (instances). Find $d \in \boldsymbol{D}$ with the lowest distance and the highest count and find the first instance/generalized rectangle found with it. Go to Step 3.
3. If the found d is correct severity compatible with the mammographic mass data, increment positive count for this instance/generalized exemplar. Otherwise, increment negative count with this instance/generalized exemplar and check if the exemplar falls inside a generalized exemplar of another $f \in \boldsymbol{D}, f \neq d$. Prune this overgeneralized exemplar if it falls there or adjust the weights for attributes with different weights if it does not fall there. Go to Step 4.
4. If the nearest neighbor of mammogram $\boldsymbol{e}$ is a point, go to Step 5. If the nearest neighbor of mammogram $\boldsymbol{e}$ is a rectangle, extend each attribute value interval to include mammogram $\boldsymbol{e}$. If the extended rectangle covers conflicting points/rectangles, restore the rectangle to its original size and keep storing mammogram $\boldsymbol{e}$. If there is no overlap there, retain modifications to the rectangle and discard mammogram $\boldsymbol{e}$. Go to Step 1.
5. Create a new rectangle that covers both the nearest neighbor and mammogram $\boldsymbol{e}$. If the new rectangle covers conflicting points/rectangles, discard the new rectangle and keep storing $\boldsymbol{e}$. If not, store the generalized exemplar corresponding to the new rectangle and discard $\boldsymbol{e}$. Go to Step 1.

The original algorithm ignores attributes with missing values. If the value of an attribute in $\boldsymbol{B}$ is missing for either the instance or the exemplar against which it is being compared, this missing value does not become a part of the distance and it is divided by the number of attributes without missing values. It is argued in this paper with experimentation that for mammographic mass data from Section II it is better to replace missing values of an attribute with the mean of non-missing values of this attribute. The amount of missing values for each attribute is less than 10 percent and so the mean value combined with the other values gives possibly better knowledge about the mammogram than no value at all. In addition, in the original algorithm, Step 2 is used for determining the severity of an unknown mammogram. Only one closest exemplar is used there. This paper argues with experimentation on the mammographic mass data from Section II that it is better to use the voting of the three closest exemplars (see Fig. 3). Number three is computed as the even number of possible severities plus one. With this, the severities of several closest exemplars are taken into consideration without too many of them and there is always one chosen severity.

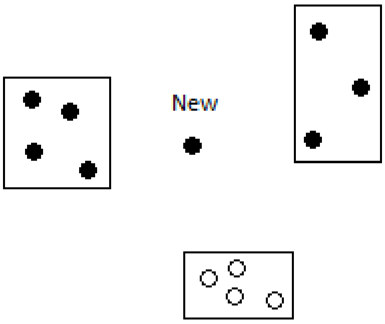


Figure 3. Voting of three closest exemplars.

IV. Experimental Results

Experiments carried out with the employed algorithm and its modifications from Section III and the mammographic mass data are described here. The presented algorithm and its modifications are written in the Java programming language. In the experiments, sensitivity and specificity were computed on the basis of created confusion matrix in 10-fold cross-validation. Sensitivity measures the ratio of malignant mammograms which are correctly identified as malignant ones. It is defined as $\frac{\mathrm{tp}}{\mathrm{tp+fn}}$ where tp (true positives) is the number of mammograms which are malignant and predicted as malignant and fn (false negatives) is the number of mammograms which are malignant and predicted as benign. Low sensitivity leads to situations which cause that dangerous masses are not treated, i.e. sensitivity is related to life-threatening situations. Specificity measures the ratio of benign mammograms which are correctly identified as benign ones. It is defined as $\frac{\mathrm{tn}}{\mathrm{tn+fp}}$ where tn (true negatives) is the number of benign mammograms which are identified as benign ones and fp (false positives) is the number of benign mammograms which are identified as malignant ones. Low specificity leads to situations which cause that benign masses are treated without any reason and thus the costs are increased and the comfort of the patients is decreased. Both of these criteria are important and so their sum is considered. Regarding negativity and positivity, $D(\boldsymbol{e})$=*benign* is considered to be negative and $D(\boldsymbol{e})$=*malignant* is considered to be positive. In the 10-fold cross-validation [9], the mammographic mass data described in Section II is randomly partitioned into 10 subgroups with equal proportions of malignant and benign mammograms. One subgroup is retained for validation of true and predicted outcomes, and the other nine subgroups are used by the algorithms as training data. In the end, each subgroup is used exactly once for validation. This is repeated 10 times, with each of the 10 created subgroups used exactly once as the validation data. A confusion matrix is also created in the beginning and it is updated with data related to all of the ten validations.

TABLE III. Experimental Results

Measure/Method	NN	NNG	NNG-Mod1	NNG-Mod2
Sensitivity	0.71	0.75	0.78	0.74
Specificity	0.80	0.73	0.78	0.83
Sum	1.51	1.48	1.56	1.57

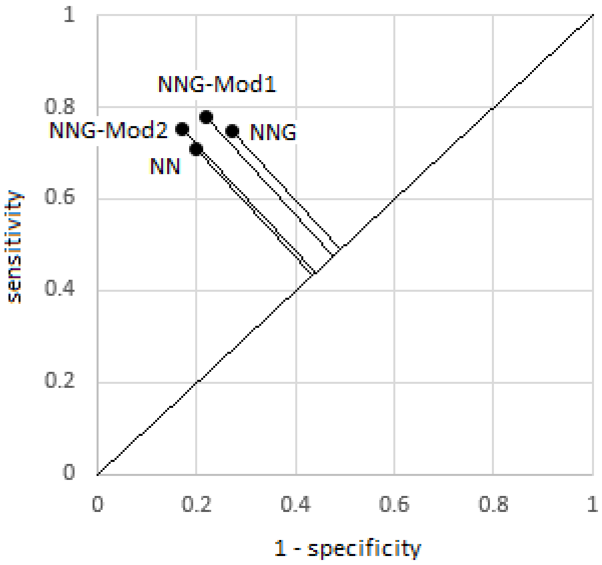


Figure 4. ROC graph.

Validation results are given in Table III where columns represent particular algorithm versions and rows represent computed measures. NN is included in addition to the analyzed algorithm and it represents the traditional nearest neighbor algorithm. Its implementation in tool Weka [13] as class IBk is used. NNG is the nearest neighbor method using non-nested generalized exemplars [3]. NNG-Mod1 is NNG with modified distance computation between attributes with missing values (see Section III). NNG-Mod2 denotes NNG with the use of several exemplars in diagnosis and with modified distance computation between attributes (see Section III). Sensitivity is sensitivity for particular algorithm versions, Specificity is specificity for particular algorithm versions, and Sum is the sum of sensitivity and specificity for particular algorithm. The results show that accuracy and specificity increased, which leads to less false positive situations with unnecessary biopsies for benign situations. The presented modifications also outperformed the traditional nearest neighbor method. The ROC curve in Fig. 4 shows some of the results presented in Table III graphically and in a way which is typical for medical specialists. Particular methods are shown as plots in the ROC space where a performance indicator for each method is the distance from the random guess line instead of the sum of sensitivity and specificity. It can be seen again in Fig. 4 that the presented modified algorithm versions outperform NNG and NN.

V. Conclusions

An improved Nearest Neighbor Method Using Non-nested Generalized Exemplars was presented so that it achieved better accuracy of predictions than the traditional nearest neighbor algorithm and keeps its advantages over it such as a much lower required amount of stored data for executing predictions and a much lower amount of potential neighbors which have to be checked. The improved algorithm was applied to the assessment of mammograms by cardiologists with the objective of avoiding unnecessary biopsies for benign situations, which is a common issue in breast cancer treatment. A distance computation between attributes with missing values was modified in the nearest neighbor method, which lead to fewer possible unnecessary biopsies for benign situations in the method. Additional use of several exemplars in diagnosis for a mammogram lead to even fewer possible unnecessary biopsies. These results were validated through 10-fold cross-validation where sensitivity, specificity and their sum were computed. The fact if dangerous masses were treated correctly was measured by sensitivity. The fact if benign masses were left without treatment was measured by specificity. With the modification of distance computation specificity increased to 0.78 and with the use of several exemplars it increased further to 0.83. The sum of sensitivity and specificity improved from 1.48 to 1.57 and outperformed the traditional nearest neighbor method whose sum was 1.51. Overall, the results showed the presented modifications were useful for avoidance of unnecessary biopsies for Nearest Neighbor Method Using Non-nested Generalized.

Acknowledgment

This paper was supported by the following project: “University Science Park of the University of Zilina – Phase II” (ITMS: 313011D13) of the The Operational Programme Research and Innovation funded by the European Regional Development Fund.

OPERAČNÝ PROGRAM VÝSKUM A INOVÁCIE

Európska únia
Európsky fond regionálneho rozvoja

It was also supported by the following faculty research grant: “User-centred Approach for Decision Support Systems” (no. FVG/2/2017) of the Faculty of Management Science and Informatics, University of Zilina, Slovakia.

The authors would also like to thank Silvia Tomancova for creating software implementations of algorithms and conducting various experiments on Mammographic Mass Database as a part of her studies leading to the completion of her final thesis under the supervision of the first author Jan Bohacik.

References

[1] American Cancer Society, Breast Cancer. : American Cancer Society, 2016.

[2] S. W. Bernard, C. P. Wild, World Cancer Report 2014. : International Agency for Research on Cancer, World Health Organization, 2014.

[3] M. Brent, Instance-based Learning: Nearest Neighbour with Generalization. Hamilton, New Zealand: University of Waikato, 1995.

[4] D. G. Dodge, J. L. Kegel, “Advances in breast cancer screening and diagnosis,” The Journal of Lancaster General Hospital, vol. 1, no. 2, pp. 47-51, 2006.

[5] M. Elter, R. Schulz-Wendtland and T. Wittenberg, “The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process,” Medical Physics 34(11), pp. 4164-4172, 2007.

[6] J. Ferlay, E. Steliarova-Foucher, J. Lortet-Tieulent, S. Rosso, J. W. W. Coebergh, H. Comber, D. Forman, F. Bray, “Cancer incidence and mortality patterns in Europe: Estimates for 40 countries in 2012,” European Journal of Cancer, vol. 49, no. 6, pp. 1374-1403, 2013.

[7] C. Lenz, D. Schmitt, “Understanding the perceptions and unmet needs of advanced breast cancer patients,” Journal fur Pharmakologie und Therapie, vol. 23, no. 4, pp. 111-115, 2014.

[8] R. Luengo-Fernandez, J. Leal, A. Gray, R. Sullivan, “Economic burden of cancer across the European Union: A population-based cost analysis,” The Lancet Oncology, vol. 14, no. 12, pp. 1165-1174, 2013.

[9] McLachlan G., Do K.-A., Ambroise C.: Analyzing Microarray Gene Expression Data. San Diego, USA: Willey, 2004.

[10] J. O’Shaughnessy, “Extending survival with chemotherapy in metastatic breast cancer,” Oncologist, vol. 10, pp. 20-29, 2005.

[11] A. A. Rao, J. Feneis, C. Lalonde, H. Ojeda-Fournier, “A pictorial review of changes in the BI-RADS fifth edition,” Breast Imaging, vol. 36, no. 3, 2016.

[12] A. L. Siu, “Screening for breast cancer: U.S. Preventive Services Task Force recommendation statement,” vol. 164, no. 4, pp. 279-297, 2016.

[13] I. H. Witten, E. Frank, M. A. Hall, Practical machine learning tools and techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufman Publishers, 2011.

[14] A. Znaor, C. Van Den Hurk, M. Primic-Zakelj, D. Agius, D. Coza, A. Demetriou, N. Dimitrova, S. Eser, H. Karaklinc, S. Zivkovic, F. Bray, J. W. W. Coebergh, “Cancer incidence and mortality patterns in South Eastern Europe in the lat decade: Gaps persist compared with the rest of Europe,” European Journal of Cancer, vol. 49, no. 7, pp. 1683-1691, 2013.

Unsupervised Method for Detection of High Severity Distresses on Asphalt Pavements

Emir Buza, Amila Akagic, Samir Omanovic, Haris Hasic
University of Sarajevo, Faculty of Electrical Engineering,
Department for Computer Science and Informatics
Zmaja od Bosne bb, Kampus Univerziteta, 71000 Sarajevo
Email: {emir.buza, amila.akagic, samir.omanovic, haris.hasic}@etf.unsa.ba

***Abstract*—Efficient detection of distresses on asphalt pavements has a great impact on safe driving, thus it has been very active research subject in recent years. High severity level distresses, such as potholes, are the most severe threat to safe driving, hence timely detecting and repairing potholes is crucial in ensuring safety and quality of driving. Existing methods often require sophisticated equipment and algorithms with high-computational pre-processing steps for analysis of substantial amount of existing data (images or videos). In this paper, a new unsupervised method for detection of high severity distresses on asphalt pavements was proposed. The method was tested on highly unstructured image data set captured from different cameras and angles, with different irregular shapes and number of potholes to demonstrate its capability. Results indicated that the method can be used for rough detection and estimation of damaged pavements.**

Keywords: Pothole detection, Unsupervised Classification, Image processing, Image segmentation, Image Clustering, Computer Vision

I. Introduction

Rapidly and accurately detecting various distresses on asphalt surface pavement is crucial for safety, serviceability, rideability and quality of driving. It is also helpful for reducing the cost of vehicle maintenance. Distresses can be categorized as high or low severity level distresses. Former are potholes and cracks, and latter are patches, surface deformations and defects. Among all, potholes are the most severe threat to safe driving. *Potholes* are defined as a small, bowl-shaped depressions in pavement surface. They represent important indication of the structural problems on asphalt pavement surfaces.

The manual visual inspection at regular intervals is currently the main form of condition assessment in most countries. Unfortunately, due to insufficient inspection and condition assessment, this method had some tragic consequences in countries such as Japan [1] and USA [2]. The other form of inspection is automatic visual inspection which accounts for only 0.4% of all inspections. Automatic inspection uses specialized vehicles that are mounted with laser scanners, pavement profilers, accelerometers, image and video cameras, and positioning systems. Automatic inspection methods have several restrictions, such as number of detectable defects, cost of sensor equipment and level of attainable details.

There are three main approaches for pothole detection available in literature: 3D reconstruction-based, vibration-based and vision-based. In this paper, we address the problem of automatic pothole detection by automatic analysis of selected 2D images of asphalt surface pavements. We use vision-based approach which is becoming more attractive since inexpensive and omni-present cameras or smartphones can be used, and it has already shown promising results. However, most methods in literature are based on supervised classification which greatly relies on knowledge of a user. For example, a method can require existence of a number of pothole texture samples for training phase. The overall performance of such a system is limited and depended on how well the system has been trained and how many samples were used.

Our method uses unsupervised classification based only on analysis of an image, hence high-computational pre-processing steps nor training data are not required. In the first step of our method, an image is prepared for region of interest (ROI) extraction by means of image segmentation and clustering. The region of interest is asphalt surface pavement, which is implied from the definition of a problem. Thus, in the second step, all possible asphalt pavement surfaces are selected by applying different selection criteria through three different cluster selection phases. Then, linear regression is applied to eliminate surfaces that are not asphalt pavement surfaces. Once the asphalt pavement is selected, potholes are detected by comparing two cropped images [3]. The accuracy of region of interest extraction is very important for overall performance of our method.

The effectiveness of our method was verified on an image data set from selected images from Google search engine. The data set contains highly unstructured images taken from different cameras and shooting angles, with different irregular shapes and number of potholes. Our method is designed to be used under daytime fair weather conditions, which is consistent with the current practice. The method is low-cost and efficient since it can use off-the-shelf inexpensive equipment, while the image resolution or quality does not affect the accuracy of detection. The results showed that the method can detect potholes with reasonable accuracy.

This paper is organized as follows: related work is briefly reviewed in Section II. In Section III, a new method for detection of high severity distresses on asphalt pavements has been proposed. The implementation details, data set and experimental results have been presented in Section IV. The

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

paper is concluded with some remarks in Section V.

II. Related Work

The existing approaches for pothole detection can be classified into three groups [4]: 3D reconstruction-based, vibration-based and vision-based. 3D reconstruction of the pavement surface can be acquired by laser scanners [5] [6], stereo-vision algorithms with a pair of video cameras [7], [8], [9] and visualization using Microsoft Kinect sensor [10], [11], [12]. These methods visualize 3D pavement surface data with sophisticated and high cost equipment, and as a consequence exhibit high computational cost. Vibration-based approaches [13], [14], [15], [16] use accelerometers to assess pavement condition based on the mechanical responses of the vehicle which carries equipment. The drawback is that they cannot be used at bridge expansion joints and cannot detect potholes in the center of a lane.

Vision-based approach uses various image processing techniques to detect potholes on 2D images or video input. In [17], Karuppuswamy et al. proposed a new method based on integrating vision and motion system for detection of simulated potholes. The application of the method is limited due to simulation limitations (potholes smaller than 2 feet in diameter, and the color of pothole). In [18], Koch et al. proposed a novel supervised approach for automated pothole detection based on asphalt pavement images. This approach is based on pothole texture extraction and comparison, where the surface texture inside a pothole candidate has to be described and compared with the texture of the surrounding region. This implies existence of a number of pothole texture samples for training phase, thus a system can detect potholes based on how well it has been trained. Additionally, such a system often requires spot filters to be applied to original gray-level image in order to emphasize structural texture characteristics. In [19], Wang et al. proposed pothole detection and segmentation method based on Wavelet energy field, which integrates the gray and texture features together to highlight the pothole region. In [3] [20], two unsupervised vision-based cost-effective methods are proposed. The first method was based on image processing and spectral clustering for identification and rough estimation of potholes, while the second on manipulation of B component in RGB color space, and two-level dynamic selection of asphalt pixels (seed points) based on standard deviation of an image. They differ in accuracy and speed of detection, as well as in level of details required to detect a pothole.

Detection of potholes over a sequence of frames has been reported in Koch et al. [21] and [22], who extended their original methods with video processing. Radopoulou et al. [23] proposed the use of Semantic Texton Forests for detection of several defects occurring in video frames. In [24], feature-based pothole detection method is proposed. It uses various features in 2D images such as the standard deviation of a candidate region, OHI, differences in the standard deviations, and averages between the outside and inside of a candidate region.

III. Methodology

The overall design of our method is shown in Figure 1. The method consists of three major steps:

(A) Image Pre-processing
(B) Region of Interest Extraction
(C) Pothole detection

Each step is described in detail in separete subsection.

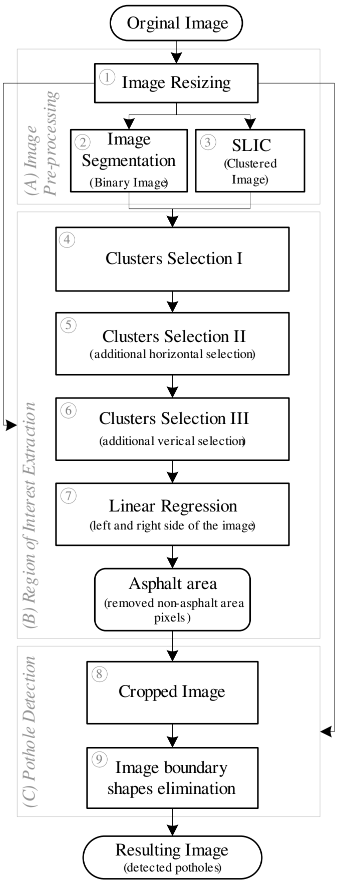
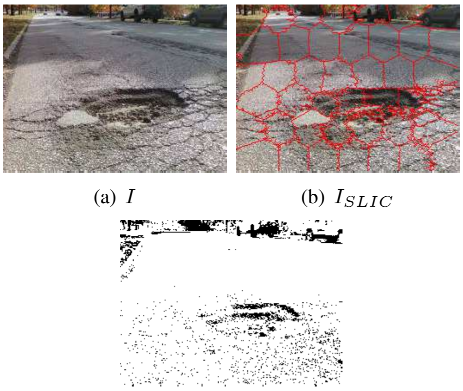

Orginal Image
(A) Image Pre-processing
1 Image Resizing
2 Image Segmentation (Binary Image)
3 SLIC (Clustered Image)
(B) Region of Interest Extraction
4 Clusters Selection I
5 Clusters Selection II (additional horizontal selection)
6 Clusters Selection III (additional verical selection)
7 Linear Regression (left and right side of the image)
Asphalt area (removed non-asphalt area pixels)
(C) Pothole Detection
8 Cropped Image
9 Image boundary shapes elimination
Resulting Image (detected potholes)

Fig. 1: Block diagram of proposed method for pothole detection.

A. *Image Pre-processing*

The Image Pre-processing step consists of three phases: image resizing, conversion to binary image and application of Simple Linear Iterative Clustering (SLIC) superpixels method [25]. In the first phase, an image is resized to 400 columns for the case when the number of columns is greater than 400 or smaller then 180. The size of an image does not effect the quality of detection, thus the number of computations is decreased by simply reseizing an image.

In the second phase, an image is converted into a binary image to obtain B component by replacing the original image pixels of R, G and B components as shown by criterias in equations (1), (2) and (3), where σ is standard deviation of an image. In the third phase, SLIC superpixels method was applied. SLIC method is an adaptation of k-means clustering



(a) I (b) I_{SLIC}

(c) $\hat{I}_{[B]}$

Fig. 2: Image Pre-processing: (a) The orginal image after resizing I; (b) The clustered image I_{SLIC} and (c) the resulting image $\hat{I}_{[B]}$ after appying (1), (2) and (3).

approach for improving overall segmentation performance. The result of applying SLIC is faster and more memory efficient implementation of image segmentation. We define weighting factor $m = 40$, since spatial proximity is more important. This enforces generation of more regular and smoother shapes superpixels. The number of superpixels k is selected by using the equation (4).

$$I'_{[R,G,B](x,y)} = \begin{cases} (0,0,0), & I_{G(x,y)} - \frac{I_{R(x,y)}+I_{B(x,y)}}{2} + \\ & 15 > 0 \\ (0,0,0), & I_{R(x,y)} - I_{B(x,y)} > 20 \\ (0,0,0), & I_{G(x,y)} - I_{R(x,y)} > 15 \end{cases} \quad (1)$$

$$I''_{[R,G,B](x,y)} = \begin{cases} (0,0,0), & I'_{[R,G,B](x,y)} > (255 - \sigma) \\ (0,0,0), & I'_{[R,G,B](x,y)} < \sigma \end{cases} \quad (2)$$

$$\hat{I}_{[R,G,B](x,y)} = \begin{cases} (255,255,255), & I''_{[R,G,B](x,y)} > 0 \\ (0,0,0), & I''_{[R,G,B](x,y)} = 0 \end{cases} \quad (3)$$

$$k = ceil(\sigma) \quad (4)$$

The results of these three phases are shown in Fig. 2 (a,b,c), respectively. Images I_{SLIC} and $\hat{I}_{[B]}$ are passed to the next step.

B. Region of Interest Extraction

The first input image $\hat{I}_{[B]}$ is black and white image with pixel values (0,255). The second input image I_{SLIC} is the result of SLIC method. In order to extract asphalt pavement surface from clustered image, clusters which contain asphalt color are selected with different selection criteria described in the first three phases: 1) Clusters selection I, 2) Clusters selection II and 3) Clusters selection III. In the final phase, linear regression is used to extract potential asphalt pavement surface.

1) Clusters selection I: Binary image $I^{[binary]}_{(x,y)}$ for each cluster C_i $(i = 1, 2, ..., k)$ is calculated with equations (5), (6), (7), where:

- $\bar{\hat{I}}_{[B](x,y)}$ is average pixel value in a cluster C_i of blue component $\hat{I}_{[B]}$ image,
- δ is threshold for $\bar{\hat{I}}_{[B](x,y)}$,
- μ is average pixel value of entire image I,
- $\mu^{[R]}_{C_i}$ is average pixel value of red component in a cluster C_i,
- $\mu^{[G]}_{C_i}$ is average pixel value of green component in a cluster C_i,
- $\mu^{[B]}_{C_i}$ is average pixel value of blue component in a cluster C_i and
- N_{C_i} and M_{C_i} represent number of rows and columns in a cluster C_i.

The region that is selected with (5) is shown in Fig. 3(a).

$$I^{[binary]}_{(x,y)} = \begin{cases} 1, & \bar{\hat{I}}_{[B](x,y)} > 255 - \delta \text{ AND} \\ & (\mu^{[G]}_{C_i} - \mu^{[R]}_{C_i} < 2 \text{ OR} \\ & \mu^{[B]}_{C_i} > \mu^{[G]}_{C_i}) \text{ AND} \\ & \sigma_{C_i} \leq 30 \\ 0, & otherwise \end{cases} \quad (5)$$

$$\delta = \begin{cases} \sigma/4, & \mu/\sigma > 4 \\ \sigma/2, & \mu/\sigma \in (2.5, 4] \\ \sigma, & \text{otherwise} \end{cases} \quad (6)$$

$$\mu^{[R]}_{C_i} = \frac{1}{N_{C_i} \times M_{C_i}} \sum_x \sum_y p(x,y)_{[R]} \quad (7)$$

2) Clusters selection II: In this phase, the region of interest was increased by subtracting the threshold value with SLIC weighting factor m. The selection of additional pixels is described in (8), where σ_{C_i} is standard deviation of a cluster C_i, $\bar{\sigma}_C$ is average value of standard deviation from the first phase, and $\bar{\mu}_{C_i}$ is average value of average pixel value of a color components in cluster C_i.

The new region was selected and added to the previously selected region and the results are shown in Fig. 3(b).

$$I^{[binary]}_{(x,y)} = \begin{cases} 1, & \bar{\hat{I}}_{[B](x,y)} > 255 - \delta - m \text{ AND} \\ & \mu^{[R,G,B]}_{C_i} > \bar{\mu}^{[R,G,B]}_{C} \text{ AND} \\ & \sigma_{C_i} \leq \bar{\sigma}_C \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$\sigma_{C_i} = \sqrt{\frac{(\mu^{[R]}_{C_i} - \bar{\mu}_{C_i})^2 + (\mu^{[G]}_{C_i} - \bar{\mu}_{C_i})^2 + (\mu^{[B]}_{C_i} - \bar{\mu}_{C_i})^2}{2}}, \quad (9)$$

where $\bar{\mu}_{C_i} = \frac{\mu^{[R]}_{C_i} + \mu^{[G]}_{C_i} + \mu^{[B]}_{C_i}}{3}$.

3) Clusters selection III: In this phase, independent regions are connected by searching for unselected clusters that are usually found between selected clusters (Cluster Selection I and II). Horizontal search is performed first and later vertical search. The result is shown in Fig. 3(c).

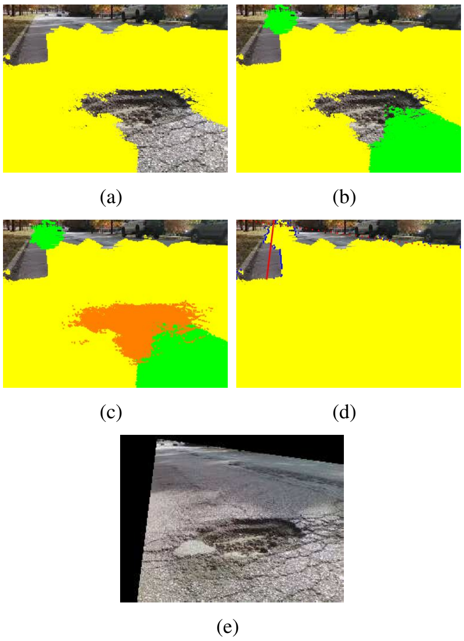


Fig. 3: Region of Interest Extraction: (a) Clusters selection I (yellow); (b) Clusters selection II (green). (c) Clusters selection III (orange), (d) Regression line of left and right side of the image (red lines), and (e) The result after applying regression line on an original image.

4) Linear Regression: Some lateral clusters selected in previous phase cover non-asphalt pixels, thus linear regression was applied in order to eliminate surfaces that are not asphalt. Left and right endpoints of each cluster were selected by searching an image from top, excluding the edges of an image. These endpoints are shown in Fig. 3(d) as blue points, while results of linear regression is shown as red lines (left and right). The result after applying linear regression is shown in Fig. 3(e).

C. Pothole detection

The third and the final step consists of two phases: 1) the method of cropped images and Otsu thresholding [3], and 2) linear and image boundary shapes elimination. The measure for removing all linear regions is defined by eccentricity ϵ of shapes [26].

The resulting image after the first phase is shown in Fig. 4(a). Based on the shapes detected, rectangles were drawn over detected potholes in the final step, as shown in Fig. 4(b).

IV. Implementation and Results

A. Implementation

The method has been implemented in MATLAB (R2010b) with Image Processing Toolbox. The performance results are meassured on Intel Core2 2.80 GHz CPU with 8GB of RAM.

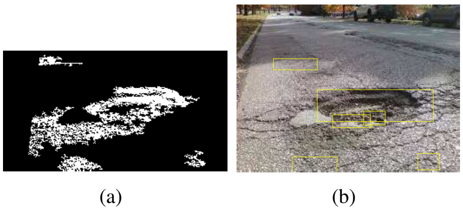


Fig. 4: Pothole detection: a) Result of cropped image; b) Distresses detected.

B. Dataset

It is common practice to test an algorithm for detection of distresses on a data set produced with in-expensive and omnipresent equipment mounted on a passenger vehicle, i.e. off-the-shelf digital cameras for video and photo acquisition. In this case, images are captured from one position and angle (usually with front or back camera). This enables extraction of parameters (such as vanishing point, pavement lane and its surroundings, etc.) that can be used to narrow the scope of object search. Unfortunately, no publicly available data set exists for detection and classification of any defect type [27]. This prevents true verification of results and makes comparison between different methods very difficult.

Our method is not bound by any specific structure of an image, position or an angle of camera, thus its effectiveness was tested on highly unstructured data set [20]. This data set consists of images taken from different types of cameras, various angles and positions, with different irregular shapes and a number of potholes. We believe that this unstructured data set can be used to examine the true potential of any pothole detection method.

C. Results

The method was tested on more than 80 different pothole images captured from various cameras, shooting angles, different irregular shapes and number of potholes, different sizes and backgrounds, and weather conditions. The capabilities of our method on selected images are shown in Fig. 5. We chose unstructured images considered difficult to segment: with or without lanes/lines, objects (cars), asphalt pavement, water/reflection, vanishing point, surrounding greenery, etc.

The detection accuracy is calculated by comparing detected number of potholes and an actual number of potholes visible on an image (calculated manually). Potholes that occur on asphalt-surfaced pavements are classified by severity of damage as: 1) *small damage* when there is a single pothole that just started forming with diameter less than 100 mm, 2) *middle damage* when there are many independent potholes with diameter between 100 and 300 mm, and 3) *high damage* when there are partly connected potholes with diameter greater or equal to 300 mm [28]. The small damage potholes are not critical for safe driving, thus detection of this type of pothole is not crucial for efficiency of our method. However, it is beneficial if a method can detect these types of potholes. Our method did not detect all small potholes, however it can detect many of them (this is demonstrated in image 7 in Fig. 5). The biggest challenge were middle damage potholes because there are many independent potholes with very board diameter specter. In this case, detection is considered successful if the method detected more than half of potholes on an image. Our method detected all high damage potholes in all tested images.

When compared to previous work [20], the method showed better accuracy under daytime fair weather conditions (in some cases more than 10%), while in cases when there is water in a pothole results were slightly worse. The region of asphalt surface pavement extracted by new method is significantly better when compared with [3]. This comes at a cost of slower execution time of one-half to one order of magnitude, which was expected, since SLIC superpixels method was introduced. The execution time of the method varied significantly depending on the complexity of an image. The average execution time on a given data set was 2489.9 ms.

V. Conclusion

In this paper a new unsupervised method for automatic detection of potholes was proposed. The effectiveness of our method depends on the extraction accuracy of ROI. The possibility of detecting pothole was decreased when ROI did not include entire asphalt surface or when ROI included area that is not asphalt. Once ROI is extracted, the potholes were detected by comparing two cropped images and performing Otsu thresholding method. Once all linear and image boundary shapes were eliminated, the remaining regions were labeled as potential potholes.

The method was tested on more than 80 different pothole images and detection accuracy was calculated manually. On a given data set, the method detected potholes with reasonable accuracy, thus it showed promising results for rough detection and estimation of damaged pavements.

In future, our research will focus on series of overlapping pictures, which will enable online detection of high severity distresses on asphalt pavements.

References

[1] Asakura, T. and Kojima, Y., 2003. *Tunnel maintenance in Japan.* Tunnelling and Underground Space Technology, 18(2), pp.161-169.

[2] National Transportation Safety Board, 2008., *Collapse of I-35W Highway Bridge, Minneapolis, Minnesota.* [Online]. Available: "http://www.dot.state.mn.us/i35wbridge/ntsb/finalreport.pdf"

[3] Buza, E., Omanovic, S. and Huseinovic, A., 2013. *Pothole detection with image processing and spectral clustering.* In Proceedings of the 2nd International Conference on Information Technology and Computer Networks (pp. 48-53).

[4] Kim, T. and Ryu, S.K., 2014. *Review and analysis of pothole detection methods.* Journal of Emerging Trends in Computing and Information Sciences, 5(8), pp. 603-608.

[5] Chang, K.T., Chang, J.R. and Liu, J.K., 2005. *Detection of pavement distresses using 3D laser scanning technology.* In Computing in Civil Engineering, pp. 1-11.

[6] Li, Q., Yao, M., Yao, X. and Xu, B., 2009. *A real-time 3D scanning system for pavement distortion inspection.* Measurement Science and Technology, 21(1), p.015702.

[7] Wang, K.C., 2004. *Challenges and feasibility for comprehensive automated survey of pavement conditions.* In Applications of Advanced Technologies in Transportation Engineering, pp. 531-536.

[8] Hou, Z., Wang, K.C. and Gong, W., 2007. *Experimentation of 3D pavement imaging through stereovision.* In International Conference on Transportation Engineering, pp. 376-381.

[9] Staniek, M., 2013. *Stereo vision techniques in the road pavement evaluation.* In XXVIII International Baltic Road Conference, pp. 1-5.

[10] Joubert, D., Tyatyantsi, A., Mphahlehle, J. and Manchidi, V., 2011. *Pothole tagging system.*

[11] Jahanshahi, M.R., Jazizadeh, F., Masri, S.F. and Becerik-Gerber, B., 2012. *Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor.* Journal of Computing in Civil Engineering, 27(6), pp. 743-754.

[12] Moazzam, I., Kamal, K., Mathavan, S., Usman, S. and Rahman, M., 2013. *Metrology and visualization of potholes using the microsoft kinect sensor.* In Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, pp. 1284-1291.

[13] Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G. and Selavo, L., 2011. *Real time pothole detection using android smartphones with accelerometers.* In Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on, pp. 1-6.

[14] Yu, B.X. and Yu, X., 2006. *Vibration-based system for pavement condition evaluation.* In Applications of Advanced Technology in Transportation, pp. 183-189.

[15] De Zoysa, K., Keppitiyagama, C., Seneviratne, G.P. and Shihan, W.W.A.T., 2007. *A public transport system based sensor network for road surface condition monitoring.* In Proceedings of the 2007 workshop on Networked systems for developing regions, pp. 9.

[16] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S. and Balakrishnan, H., 2008. *The pothole patrol: using a mobile sensor network for road surface monitoring.* In Proceedings of the 6th international conference on Mobile systems, applications, and services, pp. 29-39.

[17] Karuppuswamy, J., Selvaraj, V., Ganesh, M.M. and Hall, E.L., 2000. *Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment.* In Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision, Vol. 4197, pp. 70-81.

[18] Koch, C. and Brilakis, I., 2011. *Pothole detection in asphalt pavement images.* Advanced Engineering Informatics, 25(3), pp. 507-515.

[19] Wang, P., Hu, Y., Dai, Y. and Tian, M., 2017. *Asphalt Pavement Pothole Detection and Segmentation Based on Wavelet Energy Field.* Mathematical Problems in Engineering.

[20] Akagic, A., Buza, E. and Omanovic, S., 2017. *Pothole detection: An efficient vision based method using RGB color space image segmentation.* In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017 40th International Convention on, pp. 1104-1109.

[21] Koch, C., Jog, G.M. and Brilakis, I., 2012. *Automated pothole distress assessment using asphalt pavement video data.* Journal of Computing in Civil Engineering, 27(4), pp. 370-378.

[22] Huidrom, L., Das, L.K. and Sud, S.K., 2013. *Method for automated assessment of potholes, cracks and patches from road surface video clips.* Procedia-Social and Behavioral Sciences, pp. 312-321.

[23] Radopoulou, S.C. and Brilakis, I., 2016. *Automated detection of multiple pavement defects.* Journal of Computing in Civil Engineering, 31(2), p. 04016057.

[24] Ryu, S.K., Kim, T. and Kim, Y.R., 2015. *Feature-Based Pothole Detection in Two-Dimensional Images.* Transportation Research Record: Journal of the Transportation Research Board, (2528), pp. 9-17.

[25] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Ssstrunk, S., 2012. *SLIC superpixels compared to state-of-the-art superpixel methods.* IEEE transactions on pattern analysis and machine intelligence, 34(11), pp. 2274-2282.

[26] *Matlab Toolbox for Image Processing, Image Analysis, Region and Image Properties, http://www.mathworks.co.uk/help/images/ref/regionprops.html#bqkf8id*

[27] Koch, C., Georgieva, K., Kasireddy, V., Akinci, B. and Fieguth, P., 2015. *A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure.* Advanced Engineering Informatics, 29(2), pp. 196-210.

[28] A. Ipavec, 2012. *Study of existing standards, techniques, materials and experience with them on the European market.*, [Online]. Available: http://www.cedr.eu/download/other_public_files/research_programme/eranet_road/call_2011/design/pothole/03_pothole_d3-techniques-and-materials-study.pdf

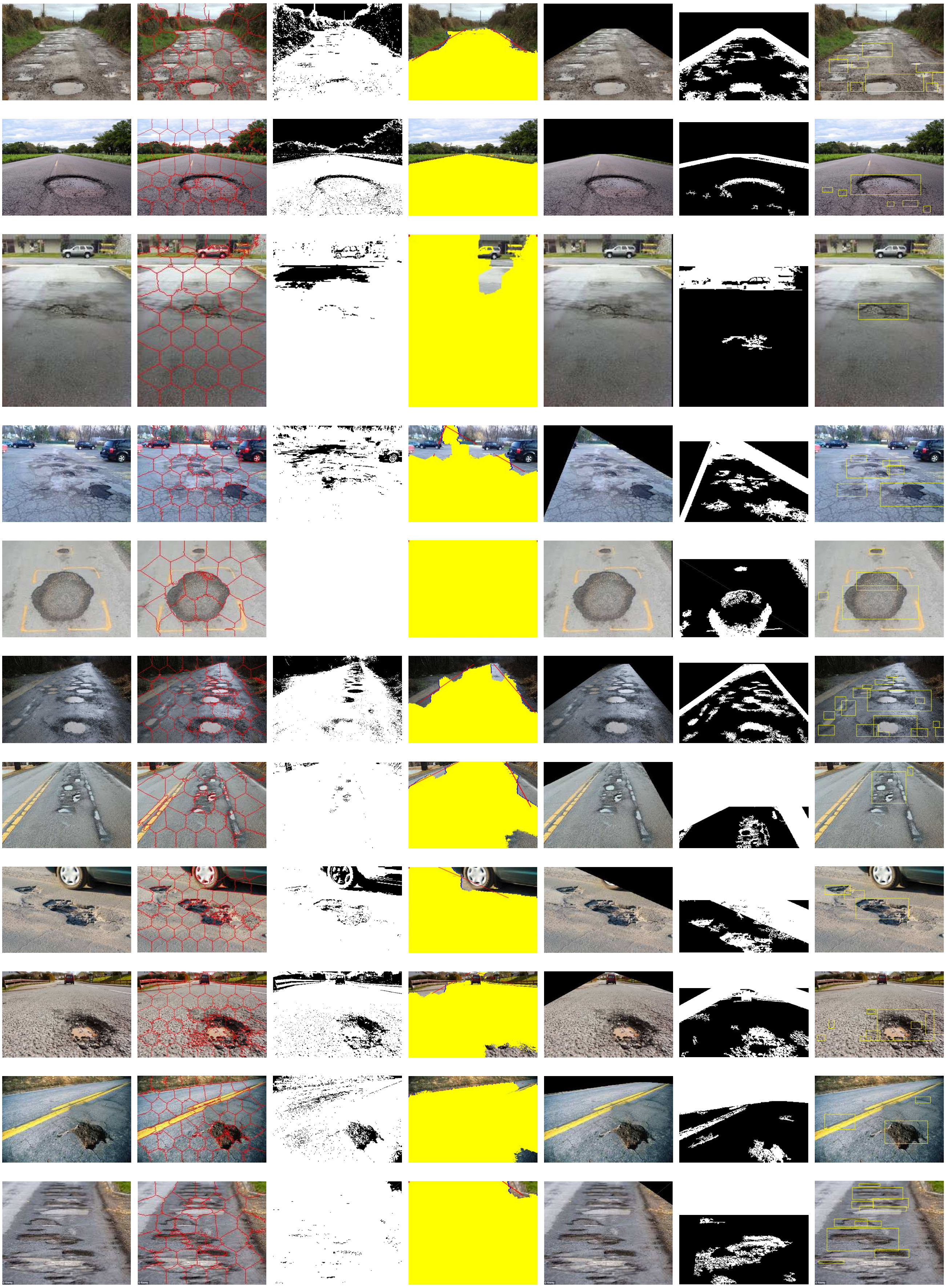


Fig. 5: Demonstration of method's capability: (1) Original image, (2) Clustered image, (3) Binary Image, (4) Clusters selection I, II, III, (5) Linear Regression, (6) Result of cropped images (7) Detected potholes.

On A Top Down Aspect Mining Approach for Monitoring Crosscutting Concerns Identification

Grigoreta-Sofia Cojocar
Department of Computer Science
Babeş-Bolyai University
1, M. Kogalniceanu Street
Cluj-Napoca, Romania
Email: grigo@cs.ubbcluj.ro

Adriana-Mihaela Guran
Department of Computer Science
Babeş-Bolyai University
1, M. Kogalniceanu Street
Cluj-Napoca, Romania
Email: adriana@cs.ubbcluj.ro

***Abstract*—*Aspect Mining* is a research domain that tries to identify crosscutting concerns implementation in software systems that were developed without using the aspect oriented paradigm. The goal is to refactor the implementation of these concerns in order to use aspects, and gain the benefits introduced by this paradigm. The aspect mining approaches proposed so far are all bottom-up approaches: starting from the source code of a software system they try to discover all the crosscutting concerns that exist in that system. In this paper we present a top-down aspect mining approach that we developed based on the observations gathered after analyzing how monitoring crosscutting concerns are implemented in three open source Java-based software systems. The approach aims to identify only the implementation of *logging* and *tracing* monitoring crosscutting concerns in Java-based software systems. We also present and discuss the results obtained by applying this approach on four open source Java software systems.**

I. Introduction

The ever increasing complexity of software systems makes designing and implementing them a difficult task. Software systems are usually composed of many different concerns. A concern is a specific requirement or consideration that must be addressed in order to satisfy the overall system. The concerns are classified in core concerns and crosscutting concerns. The core concerns capture the central functionality of a module, while crosscutting concerns capture system-level, peripheral requirements that cross multiple modules. Paradigms like procedural or object oriented programming provide good solutions for the design and implementation of core concerns, but they cannot deal properly with crosscutting concerns. Many different approaches have been proposed for the design and implementation of crosscutting concerns: subject oriented programming [1], composition filters [2], adaptive programming [3], generative programming [4], and aspect oriented programming (AOP) [5]. From these approaches, the aspect oriented programming approach has known the greatest success both in industry and academia.

For almost two decades researchers have tried to develop techniques and tools to (automatically) identify crosscutting concerns in software systems that were already developed without using AOP. This area of research is called *Aspect Mining*. The goal is to identify the crosscutting concerns, and then to refactor them to aspects, in order to obtain a system that can be easily understood, maintained and modified. In order to identify crosscutting concerns, the existing techniques try to discover one or both symptoms that appear when designing and implementing crosscutting concerns using the existing paradigms: code *scattering* and code *tangling*. Code scattering means that the code that implements a crosscutting concern is spread across the system, and code tangling means that the code that implements some concern is mixed with code from other (crosscutting) concerns.

The main contribution of this paper is to propose the first top-down aspect mining approach that tries to identify the implementation of two kinds of monitoring crosscutting concerns: *logging* and *tracing*. The approach does not aim to identify all the crosscutting concerns that exist in a software system, it only focuses on these two kinds of monitoring crosscutting concerns. We also present and discuss the results obtained by applying this approach to four real life open source Java-based software systems.

The rest of the paper is structured as follows. Section II presents an overview of the aspect mining techniques proposed so far. Section III first describes the two types of crosscutting concerns we are interested in, and then it presents the proposed approach for their automatic identification. In Section IV we present the four software systems on which we have applied the proposed approach and we discuss the results obtained. Conclusions and further work are given in Section V and Section VI, respectively.

II. Aspect Mining Techniques

The first approaches in aspect mining were query-based search techniques. The developer had to introduce a so-called seed (eg., a word, the name of a method or of a field) and the associated tool showed all the places where the seed was found. Very soon, researchers discovered that this approach to aspect mining has some important disadvantages: the user of the tool had to have an in-depth knowledge of the analyzed system, as he/she had to figure out the seed(s) to be introduced, and the large amount of time needed in order to filter the results displayed. There were many query based aspect mining tools proposed, like: Aspect Browser [6], The Aspect Mining Tool(AMT) [7], Feature Exploration and Analysis Tool(FEAT)

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

[8]. All these tools are performing the search in the source code of the mined system.

Since 2004 researchers have focused on developing aspect mining techniques that do not require an initial seed from the user. These techniques try to identify the crosscutting concerns starting just from some kind of system representation (the source code, the requirements documentation, some execution traces, etc.), and are called automated aspect mining techniques. Different approaches are used: clustering [9], [10], [11], clone detection techniques [12], [13], [14], metrics [15], association rules [16], formal concept analysis [17], [18], execution relations [19], [20], self organizing maps [21], and link analysis [22].

All the presently proposed automated aspect mining techniques try to discover all the crosscutting concerns that exist in the mined software system. The obtained results have shown that it is not so easy to develop an approach that can be used for discovering different types of crosscutting concerns. Consequently, the obtained results are not very accurate, and only some types of crosscutting concerns are discovered. If the techniques proposed in the beginning used very different approaches, the last ones are more an improvement of some of the previously proposed techniques. Even so, the results obtained by the new aspect mining techniques did not improve significantly. They obtained better results, but not much better. Also, practice has shown that not all crosscutting concerns can be refactored to aspects.

Mens et al. have conducted an analysis of the problems the proposed aspect mining techniques were encountering [23]. The main problems identified were: poor precision, poor recall, subjectivity, scalability, lack of empirical validation. The study was conducted in 2008 and since then the results obtained by the proposed aspect mining techniques did not improve much.

III. Top-Down Approach

In this section we describe our approach for identifying *logging* and *tracing* monitoring crosscutting concerns.

A. Monitoring Crosscutting Concerns

Monitoring concerns record the behaviour of a software system during development, testing and execution in its own environment. The most commonly used are: logging, tracing and performance monitoring:

- *Logging* produces messages specific to the logic carried by a piece of code.
- *Tracing* produces messages for lower-level events such as: the entry or exit of a method, exception handling or object construction, and state modification.
- *Performance monitoring* measures the time taken by specific parts of the system and/or the number of times a particular method is invoked.

It is well-known that *tracing* and *performance monitoring* are better implemented using AOP. The AOP-based solution is clearly separated from the rest of the system, can be easily understood and maintained, and it can be easily plugged-in or plugged-out of the system. As for *logging* it is not clear yet if an AOP-based solution can be designed, and if it is better than the non-AOP one.

B. Our Approach

The proposed approach is based on the results obtained in a previous study where we have manually analyzed three Java-based software systems to determine if a pattern (or patterns) can be extracted for monitoring crosscutting concerns [24]. The obtained results have shown that the most similar to a pattern is the declaration of an attribute corresponding to the object used for recording the produced messages, and then calling different methods on it. This object is often called a *logger*. This attribute is in most cases a `static` and/or `final` one (for two of the analyzed systems the percentage is higher than 90%, and for all three case studies the percentage is higher than 60%).

Listing 1 shows a fragment from the source code of `AjpMessage` class from Tomcat v9 [25] (one of the analyzed software systems).

```
package org.apache.coyote.ajp;

import org.apache.juli.logging.*;
import org.apache.tomcat.util.res.StringManager;

public class AjpMessage {

  //The logger object
  private static final Log log =
    LogFactory.getLog(AjpMessage.class);

  // The string manager for this package.
  protected static final StringManager sm =
    StringManager.getManager(AjpMessage.class);

  //Write a MessageBytes out at the
  //current write position.
  public void appendBytes(MessageBytes mb) {
    if (mb == null) {
         log.error(sm.getString("ajpmessage.
           null"), new NullPointerException());
          appendInt(0);
          appendByte(0);
          return;
     }
     // other business logic code
     appendByteChunk(mb.getByteChunk());
  }

  //Write a ByteChunk out at the
  //current write position.
  public void appendByteChunk(ByteChunk bc) {
    if (bc == null) {
         log.error(sm.getString("ajpmessage.
           null"), new NullPointerException());
         appendInt(0);
         appendByte(0);
         return;
        }
     appendBytes(bc.getBytes(), bc.getStart(),
         bc.getLength());
  }

   //other attributes and methods
}
```

Listing 1. Fragment from AjpMessage class from Tomcat v9 [25].

The fragment includes the declaration of the *logger* object (named `log`) as a `static` and `final` attribute. The object is later used in some of the methods to record the corresponding messages. This pattern was identified in most of the classes that recorded messages. Still, there are also a few exceptions to this pattern: the *logger* object is declared in a base class and methods from subclasses only use it without declaring a new attribute, or there are a few cases in which the *logger* object is declared as a local variable in the methods that needed to record messages.

Based on these results we have developed a top-down aspect mining approach that tries to identify *logging* and *tracing* monitoring crosscutting concerns by analyzing the `static` or `final` attributes defined in a Java-based software systems. The approach consists of first automatically identifying the type of the *logger* object, and then the automatic identification of the affected classes (the classes in which the monitoring concerns are implemented). Having identified the classes, then we can (automatically) analyze them to determine whether the concerns can be (automatically) refactored to aspects.

Our approach for identifying the type of the *logger* object consists of the following steps:

1) **Instrumentation.** In order to determine the `static` or `final` attributes defined in a Java-based software system we need to automatically analyze the source code (`.java` files) or the bytecode (`.class` files) of the system. The existing libraries and frameworks that allow us to analyze them (like Soot [26] or Spoon [27]) require and/or use a classpath variable that must be properly set in order to be able to analyze the input (source code or bytecode). During this step we determine all the dependencies (usually other `.jar` files) of the system to be analyzed. This is the most time consuming step, as large software systems may dependent on many different libraries that must be identified by the user if there is no additional information present (such as a Gradle [28] or Maven [29] build file).
2) **Analysis.** After the completion of the first step, we automatically identify all the `static` or `final` attributes defined in the analyzed software system. During this step we gather the following information: the type of the attribute, the number of times this type was used for declaring a `static` or `final` attribute, and the number of distinct classes in which this kind of attributes were defined. We consider that the number of distinct classes in which `static` or `final` attributes of the same type were defined is important, as in the same class many different `static` or `final` attributes of the same type may be defined.
3) **Filtering.** From the results obtained at the previous step we remove the following types: all Java primitive types (`byte`, `short`, `int`, `float`, `double`, `char`), any arrays of a primitive type (like `byte[]` or `int[][]`), all the types defined in `java.util` or `java.lang` packages (such as `java.lang.String`, `java.util.ArrayList`) but not the subpackages (types like `java.lang.reflect.Method` will not be removed), any arrays of a type defined in these two packages (eg. `java.lang.String[]`). During this step we also remove the types that were used for declaring `static` or `final` attributes in less than 3 classes. We consider that if a `static` or `final` attribute of the same type is defined in more than 3 classes than it can be considered as crosscutting, otherwise it can be considered as coupling between the corresponding types (the type of the attribute(s) and the classes in which the `static` or `final` attribute(s) was (were) defined).
4) **Sorting.** The remaining `static` or `final` attributes' types are sorted descending by the number of declaring classes. The first n results will be presented to the user as possible results for the *logger* object's type. From our observations of the manually analyzed software systems the type should be among the first ranked results. The value of n can be decided by the user (or it could have a default value).

After identifying the type of the *logger* object, we consider that *logging* and *tracing* monitoring crosscutting concerns are implemented in all the classes having an attribute of this type. These classes are determined during the analysis of all the attributes defined in the software system, so no additional computation is needed.

IV. Study

A. Case Studies

In order to verify the applicability of the previously described approach we have used four open source Java-based software systems as case studies: Spoon, Tomcat v9, Spring Framework, and ArgoUML:

- **Spoon** is an open-source library that enables transformation and analysis of Java source code. It provides a metamodel where any kind of program element such as a class, a method, a field, a statement, etc. can be accessed for reading and/or modification. The code used for our analysis was downloaded from [30].
- **ArgoUML** is an open source UML modeling tool that includes support for all standard UML 1.4 diagrams. It runs on any Java platform. We have used version 0.34 for our analysis, and the source was downloaded from [31].
- **Apache Tomcat** is an open-source web container for Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies [32]. We analyzed version 9, the source code being downloaded from [33].
- **Spring Framework** is a modular framework that helps developing Java enterprise applications by providing a comprehensive programming and configuration model [34]. The developers focus on the application-level business logic, and the framework helps putting together the final system. The source code that we have used for our analysis was downloaded from [35].

Two of these case studies (Tomcat v9 and Spring Framework) were among the systems considered for our previous

TABLE I
CASE STUDIES

Case study	Number of classes and interfaces analyzed (CI)
Spoon	724
ArgoUML	2247
Tomcat v9	2502
Spring	5235

manual analysis, but the other two were not considered in our previous study from [24]. Table I presents the number of classes and interfaces analyzed for each case study.

For the automatic analysis step we have used Soot [26], a Java optimization framework that provides various representations for analyzing and transforming Java bytecode. In order to obtain the bytecode of the systems for which we only had the source code, we have first built the systems by following the instructions described on the corresponding websites.

B. Results

Table II presents the results obtained after executing the **Analysis** and **Filtering** steps of our approach. As the results show the number of `static` or `final` attributes defined in a large Java-based software systems is big. In all four case studies the number is greater than the number of classes and interfaces analyzed, and for ArgoUML and Tomcat the number is even bigger than the doubled number of classes and interfaces. However, if we consider only the attributes' type their number decreases significantly. In all cases, the number of different types is less than 25% of the number of `static` or `final` attributes. In three of the cases (ArgoUML, Tomcat V9, and Spring) the percentage of different types over the number of `static` or `final` attributes is even less than 16%, and in the case of ArgoUML it is even less than 8% meaning that only a small part of this information is necessary for the identification of the *logger* object type. The results also show that after the **Filtering** step more than 80% of the types are removed, meaning that they are either types which are commonly used for the business logic of a system (like the primitive types or the types from the `java.util` or `java.lang` packages) or they are not crosscutting (they were used in at most 2 different classes). For all four case studies the number of possible types for the *logger* object is less than 20% of the total number of types used for declaring `static` or `final` attributes, and it is less than 4% of the total number of `static` or `final` attributes defined in the system.

In Table III is presented a subset of the results obtained after executing the **Sorting** step for each case study. Column **DA** represents the number of times a `static` or `final` attribute of the corresponding type was declared, **DC** represents the number of classes that declare a `static` or `final` attribute of this type, **TDC** represents the number of classes that declare an attribute of this type (independently of the modifier used: `static`, `final` or none), **TDC/CI** represents the ratio of the number of declaring classes over the total number of classes and interfaces from the system, **CCC** represents the number of classes from the software system in which the concerns are implemented (the number is determined during the manual analysis of the system), and **ACC** represents the accuracy of our approach. The accuracy is considered to be the percentage of classes from the software system that are part of the crosscutting concerns implementation and were identified as such by our approach. For this study the accuracy is computed as the percentage of **TDC/CCC**.

For the three larger case studies (ArgoUML, Tomcat v9, and Spring) the type ranked at the first position is the type of the *logger* object, showing that this approach could be used for automatic identification of *logging* and *tracing* monitoring crosscutting concerns. For these three case studies the percentage of **TDC/CI** of the type ranked at the first position is also significantly greater than the percentage of the type ranked at the second position. The results also show that the type used for the *logger* object is different for all three case studies, meaning that it is dependent on the software system.

In the case of Spoon case study the top 3 ranked types do not include the type of the *logger* object. The most used type for declaring `static` or `final` attributes is `ReplacementVisitor` from `spoon.support.visitor.replace` package which was used in more than 10% of the classes from system. In this case the type of the *logger* object was ranked only at the position 15, being declared as a `static` or `final` attribute in only 4 classes, meaning less than 1% of the total number of classes. A more in-depth analysis has determined that the type was actually used in 5 classes, but in one class the *logger* object was not declared neither as a `static` nor a `final` attribute.

The study presented in [24] has also shown that there are software systems which use more than one type for the *logger* object. The manual analysis of two of the systems used in this study have shown that they use more than one type for the *logger* object: Tomcat v9 uses 2 types, and Spring Framework uses 3 types. For Tomcat case study the results obtained by our approach actually included both types, but the second type was ranked at the position 96 (from 136 possible positions), as shown in Table III. The second type was used only in 4 classes, and the *logger* object was declared as a `static` or `final` attribute in only 3 of the classes. In the case of Spring Framework, only one type was given among the possible results. The other two types were not included because they were used in at most 4 classes, and in some of these classes the *logger* object was not defined as a `static` or `final` attribute.

As shown in Table III, the accuracy of our approach is higher than 75% for all the three larger case studies, for two of them the accuracy being even higher than 90%. Even if for Spoon case study, where the type of the *logger* object is not among the top 3 ranked possible types, after choosing the correct type, the accuracy of our approach is 100%.

TABLE II
CASE STUDIES RESULTS - ANALYSIS AND FILTERING STEPS

Case study	Number of `static` or `final` attributes	Different types (DT)	Percentage of DT/attributes	Number of types after filtering (TF)	Percentage of TF/DT	Percentage of TF/attributes
Spoon	879	213	24.23%	35	16.43%	3.98%
ArgoUML	5374	394	7.33%	57	14.35%	1.06%
Tomcat v9	6630	733	11.05%	136	18.55%	2.05%
Spring	7776	1198	15.40%	214	17.86%	2.75%

TABLE III
CASE STUDIES RESULTS -SORTING STEP

Case study	Rank	Attribute's type	DA	DC	TDC	TDC/CI	CCC	ACC
ArgoUML	**1**	**org.apache.log4j.Logger**	**253**	**253**	**253**	**11.25%**	**324**	**78.08%**
	2	org.argouml.language.java.reveng.JavaParser	89	89	89	3.96%	-	-
	3	org.argouml.configuration.ConfigurationKey	93	23	24	1.06%	-	-
Tomcat v9	**1**	**org.apache.juli.logging.Log**	**274**	**274**	**276**	**11.03%**	**288**	**95.83%**
	2	org.apache.tomcat.util.res.StringManager	188	187	187	7.47%	-	-
	3	org.apache.catalina.tribes.util.StringManager	37	37	37	1.47%	-	-
	...	...	...	...	...	...	-	-
	96	**java.util.logging.Logger**	**4**	**3**	**4**	**0.15%**	**4**	**100%**
Spring	**1**	**org.apache.commons.logging.Log**	**357**	**355**	**365**	**6.97%**	**402**	**90.76%**
	2	java.lang.reflect.Method	117	76	94	1.79%	-	-
	3	org.springframework.orm.hibernate3.HibernateTemplate	39	39	40	0.76%	-	-
Spoon	1	spoon.support.visitor.replace.ReplacementVisitor	82	82	82	11.32%	-	-
	2	spoon.reflect.factory.Factory	7	7	22	3.03%	-	-
	3	spoon.reflect.visitor.chain.CtQueryImpl	7	7	7	0.96%	-	-
	...	...	...	...	...	...	-	-
	15	**org.apache.log4j.Logger**	**4**	**4**	**5**	**0.69%**	**5**	**100%**

V. CONCLUSIONS

The conclusions that can be drawn from the results obtained in Section IV are:

- The proposed approach can be used for the identification of *logging* and *tracing* monitoring crosscutting concerns. The results obtained for the three larger case studies have shown that the analysis of `static` or `final` attributes of a software system is a good starting point for the identification of the type of the *logger* object used for these crosscutting concerns implementation. The set of classes in which an attribute of this type is defined is also a good starting point for determining all the affected parts of the software systems. On these classes we can perform a more in-depth analysis in order to determine, for example, the methods in which the attribute is used and to also determine if the implementation can be refactored to aspects. As Table III shows the searching space is significantly reduced, less than 15% of the total number of classes and interfaces need to be considered for the in-depth analysis.
- This approach is *scalable*. As the results of the study have shown, the number of types considered as possible results are less than 6% of the total number of classes and interfaces. Even for large or very large software systems, the possible results for the *logger* object's type are reduced significantly. The time needed to obtain all the possible types is also small. It takes less than 3 seconds to obtain the possible types for any of our case studies. Identifying all the other parts of the concerns implementation may take longer, but it should still be an acceptable amount of type.
- The majority of the already proposed aspect mining techniques try to identify crosscutting concerns by analyzing the methods defined in a software system. However, the results obtained show that for *logging* and *tracing* monitoring crosscutting concerns, a different granularity provides more accurate results.
- The results of this approach can be considered as input for automatically refactoring the implementation of these monitoring crosscutting concerns into aspects. This approach can also be used to determine if the implementation can actually be refactored into aspects. In the analysis described in [24] we have determine that at least 25% of the messages recorded using the *logger* object are constructed using local variables. This kind of monitoring messages cannot be refactored to aspects.

We did not include in this paper a comparison of our approach with other already proposed approaches as it is difficult to compare them due to their different granularities. Also, they do not automatically separate the results based on the kind of crosscutting concerns, letting the user decide which results belongs to which crosscutting concerns.

VI. Further Work

In this paper we have presented a top-down approach for automatic identification of *logging* and *tracing* monitoring crosscutting concerns implementation. The approach analyzes the attributes defined in a Java-based software system in order to determine the *logger* object's type, and then determines all the affected classses. We have used the proposed approach on four open source Java-based software systems.

Further work will be done in the following directions:

- To apply the proposed approach on other open source (larger) case studies.
- To determine if the proposed approach can be used for software systems developed using other programming languages like C# or C++.
- To develop a plugin for a popular IDE like IntelliJ or Eclipse that will allow the automatic identification of *logging* and *tracing* monitoring crosscutting concerns.
- To determine if refactoring is possible for the analyzed software systems.
- To evaluate if the structure of the system would improve if refactoring to aspects is possible.

References

[1] W. Harrison and H. Ossher, "Subject-Oriented Programming: A Critique of Pure Objects," in *Proceedings of the Eighth Annual Conference on Object-oriented Programming Systems, Languages, and Applications*, ser. OOPSLA '93, New York, NY, USA, 1993, pp. 411–428.

[2] M. Aksit, "On the design of the object oriented language sina," Ph.D. dissertation, Department of Computer Science, University of Twente, The Netherlands, 1989.

[3] K. J. Lieberherr, "Component Enhancement: An Adaptive Reusability Mechanism for Groups of Collaborating Classes," in *Information Processing '92, 12th World Computer Congress*, J. van Leeuwen, Ed. Madrid, Spain: Elsevier, 1992, pp. 179–185.

[4] K. Czarnecki and U. Eisenecker, *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, 2000.

[5] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin, "Aspect-Oriented Programming," in *Proceedings European Conference on Object-Oriented Programming*, vol. LNCS 1241. Springer-Verlag, 1997, pp. 220–242.

[6] W. G. Griswold, Y. Kato, and J. J. Yuan, "AspectBrowser: Tool Support for Managing Dispersed Aspects," UCSD, Tech. Rep. CS1999-0640, March 2000.

[7] J. Hannemann and G. Kiczales, "Overcoming the Prevalent Decomposition of Legacy Code," in *Advanced Separation of Concerns Workshop,at the International Conference on Software Engineering (ICSE)*, May 2001.

[8] M. P. Robillard and G. C. Murphy, "Concern Graphs: Finding and Describing Concerns Using Structural Program Dependencies," in *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, 2002, pp. 406–416.

[9] D. Shepherd and L. Pollock, "Interfaces, Aspects, and Views," in *Proceedings of Linking Aspect Technology and Evolution Workshop(LATE 2005)*, March 2005.

[10] G. S. Moldovan and G. Serban, "Aspect Mining using a Vector-Space Model Based Clustering Approach," in *Proceedings of Linking Aspect Technology and Evolution (LATE) Workshop*. Bonn, Germany: AOSD'06, March, 20 2006, pp. 36–40.

[11] L. He and H. Bai, "Aspect Mining using Clustering and Association Rule Method," *International Journal of Computer Science and Network Security*, vol. 6, no. 2, pp. 247–251, February 2006.

[12] D. Shepherd, E. Gibson, and L. Pollock, "Design and Evaluation of an Automated Aspect Mining Tool," in *2004 International Conference on Software Engineering and Practice*. IEEE, June 2004, pp. 601–607.

[13] M. Bruntink, A. van Deursen, R. van Engelen, and T. Tourwé, "On the use of clone detection for identifying crosscutting concern code," *IEEE Transactions on Software Engineering*, vol. 31, no. 10, pp. 804–818, 2005.

[14] O. A. M. Morales, "Aspect Mining Using Clone Detection," Master's thesis, Delft University of Technology, The Netherlands, August 2004.

[15] M. Marin, A. van, Deursen, and L. Moonen, "Identifying Aspects Using Fan-in Analysis," in *Proceedings of the 11th Working Conference on Reverse Engineering (WCRE2004)*. IEEE Computer Society, 2004, pp. 132–141.

[16] S. Vidal, E. S. Abait, C. Marcos, S. Casas, and J. A. Díaz Pace, "Aspect mining meets rule-based refactoring," in *Proceedings of the 1st Workshop on Linking Aspect Technology and Evolution*, ser. PLATE '09. New York, NY, USA: ACM, 2009, pp. 23–27.

[17] P. Tonella and M. Ceccato, "Aspect Mining through the Formal Concept Analysis of Execution Traces," in *Proceedings of the IEEE Eleventh Working Conference on Reverse Engineering (WCRE 2004)*, November 2004, pp. 112–121.

[18] T. Tourwé and K. Mens, "Mining Aspectual Views using Formal Concept Analysis," in *SCAM '04: Proceedings of the Source Code Analysis and Manipulation, Fourth IEEE International Workshop on (SCAM'04)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 97–106.

[19] S. Breu and J. Krinke, "Aspect Mining Using Event Traces," in *Proceedings of International Conference on Automated Software Engineering (ASE)*, 2004, pp. 310–315.

[20] J. Krinke and S. Breu, "Control-Flow-Graph-Based Aspect Mining," in *Workshop on Aspect Reverse Engineering (WARE)*, 2004.

[21] S. G. Maisikeli, "Aspect mining using self-organizing maps with method level dynamic software metrics as input vectors," Ph.D. dissertation, 2009.

[22] J. Huang, Y. Lu, and J. Yang, "Aspect mining using link analysis," in *Proceedings of the 2010 Fifth International Conference on Frontier of Computer Science and Technology*. IEEE Computer Society, 2010, pp. 312–317.

[23] K. Mens, A. Kellens, and J. Krinke, "Pitfalls in Aspect Mining," in *Proceedings of the 2008 15th Working Conference on Reverse Engineering*, ser. WCRE '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 113–122.

[24] G. S. Cojocar, "On Top-Down Aspect Mining for Monitoring Techniques Implementation," in *Proceedings of IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2016, pp. 249–254.

[25] "Apache Tomcat," http://tomcat.apache.org/.

[26] "Soot: a Java Optimization Framework," http://www.sable.mcgill.ca/soot/.

[27] R. Pawlak, M. Monperrus, N. Petitprez, C. Noguera, and L. Seinturier, "Spoon: A Library for Implementing Analyses and Transformations of Java Source Code," *Software: Practice and Experience*, vol. 46, pp. 1155–1179, 2015.

[28] "Gradle Build Tool," https://gradle.org/, Retrieved September, 2017.

[29] "Apache Maven Tool," https://maven.apache.org/, Retrieved September, 2017.

[30] "Spoon source code," https://github.com/INRIA/spoon, Retrieved July, 2017.

[31] "Argouml source code," http://argouml-downloads.tigris.org/nonav/argouml-0.34/ArgoUML-0.34-src.tar.gz, Retrieved March, 2016.

[32] "Apache tomcat," http://tomcat.apache.org/index.html, Retrieved January, 2016.

[33] "Apache tomcat v9," http://tomcat.apache.org/download-90.cgi, Retrieved January, 2016.

[34] "Spring framework," https://spring.io/, Retrieved January, 2016.

[35] "Spring framework source code," https://github.com/spring-projects/spring-framework, Retrieved January, 2016.

Heuristic sampling for the subgraph isomorphism problem

Uroš Čibej, Jurij Mihelič
Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
{uros.cibej, jurij.mihelic}@fri.uni-lj.si

***Abstract*—Subgraph isomorphism is one of the fundamental search problems in computer science. In this article we consider the counting variation of this problem. The task is to count all instances of the pattern G occurring in a (usually larger) graph H. All algorithms for this problem use a variation of backtracking. Most commonly they assign one vertex of G to one vertex of H at each level of the search tree. The order of vertices for the assignment is the crucial factor determining the size of this search tree. But it is very hard to determine in advance the impact of the order for a particular instance. We use a method called heuristic sampling to estimate the size of the tree. We use this estimation to select the most suitable order of vertices of G which minimizes the expected tree size. This approach is empirically evaluated on a set of instances, showing the practical potential of the method.**

I. Introduction

Backtracking algorithms are one of the most common approaches to solving hard optimization, enumeration, and decision problems. They are simple to implement and understand, but their behavior is very difficult to predict.

One of the fundamental problems classically solved by variations of backtracking algorithms is the subgraph isomorphism problem. The goal of this problem is to find a pattern graph G in a target graph H. Many variations of algorithms for this problem exist, their main differences being the methods for pruning (reducing the size) of the search tree. These variations can have a vastly different behavior on the same instance, e.g. a certain instance might be trivial for one algorithm and impossible for another. This opens up another algorithmic possibility, namely, for each instance the best algorithm could be chosen in advance, if only we knew its behavior before actually running it on the instance. Ideally, this behavior could be predicted analytically, by simply computing a set of features of the problem instance. However, this has been a difficult task, because of the large number of parameters that influence the outcome.

But luckily, there has been an interesting method developed to tackle this problem from a more pragmatical, empirical point of view. The basic method was developed by Donald Knuth [13], where he explored the possibilities to predict the search tree sizes for a few simple problems (such as the knights tour and the Instant Insanity game). The weakness of Knuth's approach is the fact that it assumes a rather homogeneous tree, i.e. all (most) of the subtrees of the search tree are somewhat similar in shape (and size). This is true for such simple problems as Knuth was dealing with, but it is not true for more complex problems.

This is why a more complex method was devised, named heuristic sampling [5]. It is a generalization of Knuth's method, where for a good implementation some specific knowledge of the underlying problem has to be provided in the form of a heuristic function.

The main goal of this article is to implement the heuristic sampling method for the subgraph isomorphism search, devise a suitable heuristic sampling function and evaluate it on a set of backtracking algorithms. We will focus on its predictive strength for choosing the best algorithm for a particular instance of the problem.

The remainder of the paper is organized as follows. Next section introduces the subgraph isomorphism problem, the notation used throughout the paper, and the algorithmic framework that we will be using in the evaluation of the proposed method. Section 3 describes the foundations of search-tree size estimation and its extension, the heuristic sampling. Section 4 describes the empirical results, namely the test set of graphs used in the evaluation and the setup of the experiments, and the analysis of the obtained results. Section 5 explores the possibilities of extension of this work into practical solvers and concludes the paper.

II. Subgraph isomorphism problem

From the viewpoint of pattern analysis and recognition, matching graphs and subgraphs is one of the most important tasks in graph processing. One approach to modeling this task is via the *subgraph isomorphism problem*. Given two graphs, namely, a *pattern graph* and a *target graph*, the problem is defined as finding a subgraph (corresponding to the pattern graph) in the target graph.

Formally, the problem can be defined as follows.

Definition 1: A *subgraph isomorphism* between a pattern $G = \langle V, E\rangle$ and a target $H = \langle U, F\rangle$ is an injective function $f : V \longrightarrow U$ satisfying the condition $(i, j) \in E \Rightarrow (f(i), f(j)) \in F$. Similarly an *induced subgraph isomorphism* between a pattern $G = \langle V, E\rangle$ and a target $H = \langle U, F\rangle$ is an injective function $f : V \longrightarrow U$ satisfying the condition $(i, j) \in E \Leftrightarrow (f(i), f(j)) \in F$.

From the algorithmic point of view it does not matter if we address the induced or the regular subgraph

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

isomorphism. All the algorithms can find both types of isomorphisms with only minor modifications.

Applications of pattern analysis using the subgraph isomorphism problem abound in various fields, such as chemistry [1], [2], social network analysis [12], and computer vision [14] . Moreover, in these domains graph databases are replacing the traditional relational databases [3]. For an extensive review of graph matching for pattern recognition see [7].

Unfortunately, the subgraph isomorphism problem is computationally very difficult, as it has been shown to be NP-complete [8]. Being such a fundamental problem, there are many attempts to overcome this theoretical barrier, and the state-of-the-art algorithms can easily solve non-trivial problems.

A. Algorithms

Since this is one of the fundamental problem in computer science, many algorithms have been proposed to solve it. The oldest algorithm that is still a reference for newer approaches in Ullmanns' algorithm [16]. For many years it was also considered the best for the problem at hand. But it was also considered to be of more theoretical nature, and there were not many attempts to engineer this backtracking into faster algorithms.

Because computers got much faster, and the applications started to deal with more data organized as graphs, the interest for this problem has been increasing, especially in the pattern matching community. New and improved algorithms have started to emerge [9], [6], [15], [17], showing remarkable results on larger and larger instances.

Even though these algorithm utilize very different techniques, all of them follow the same basic pattern of a backtrack search. We can summarize this pattern as a general framework, and we will use it as the basis for our experiments The framework is given in Algorithm 1 and can be described as follows.

Besides the input pattern G and target H, the input to this algorithm is an ordering of vertices in G and the current (partial) assignments (iso) which grows and shrinks as the algorithm traverses the search tree. At each step (node of the search tree), the vertex v is the active vertex. This vertex is defined as the first vertex in the current order. The function $getCandidates$ retrieves vertices from H which have the same connectivity in H as v has in G, based on the already assigned vertices in iso. When we reach the leaf of a search tree, the set of candidates is exactly the number of subgraph isomorphisms at that position. If we are not at the leaf we try to assign v to every possible candidate and proceed recursively.

As mentioned earlier, the most influential factor for the speed of the algorithms is the order in which vertices of G are mapped onto H. This impact of this parameter is the easiest to see in the Ullmanns' algorithm, where the vertices of G are simply ordered by their degree (in descending order). If we would to use a random order instead, even small instances become impossible to solve. So the impact of using better orders is felt much more than of any other pruning technique in this algorithm. Other algorithms use more sophisticated orders, usually taking into account also the neighborhood of the currently chosen vertex and other parameters, but the impact on the tree size is similar than with the Ullmanns' algorithm..

Algorithm 1: countIso() - counting subgraph isomorphisms with a basic backtracking algorithm.

```
Function countIso(G,H, iso, order);
v = head(order);
candidates = getCandidates(v,iso,G,H);
if size(order) = 1 then
    return $size(candidates)$;
end
nIso = 0;
forall $u \in candidates$ do
    iso = $iso \cup (v \rightarrow u)$;
    nIso = nIso + countIso(G,H, iso, tail(order));
    iso = $iso \setminus (v \rightarrow u)$;
end
return nIso
```

Since this is a very impactful technique for reducing the search-tree size, there have been also more in-depth research done, such as in [4], [18]. We will not handle all the available orders but instead focus on a subset of them, since our goal is not to exhaustively evaluate various orders but instead test how good can we predict the performance of different orders.

From a large set of available orders we chose three basic orders, which utilize very local information, and two orders which use more information about the structure of the graph. The three basic orders are as follows.

Degree (DEG) The order which was already proposed by Ullmann in [16] is the ordering of the pattern vertices by their degree (descending). The logic behind this ordering is that the nodes with the highest degree can usually be mapped to the fewest goal nodes, which reduces the search space already at the top levels of the search tree.

Depth-first search order (DFS) This order follows from a simple traversal of the tree in a depth-first manner. The first vertex is always the one with the highest degree. The trace of visited nodes gives the final order.

Breadth-first search order (BFS) Same logic as the DFS order, but with a breadth-first search.

The two more advanced orders can be viewed as an extendion of BFS. They start with the highest degree vertex and put the neighbourhood in the set of active vertices. This set is used to choose the next vertex based on a specific feature. Once a vertex is chosen, all its neighbors are added to the set of active vertices. This process is repeated until the set of active vertices is empty. The criteria by which the vertex is chosen from the active set defines the order more specifically.

Subdegree (SUB-DEG) The *subdegree* of a vertex is the number of edges from this vertex to the nodes

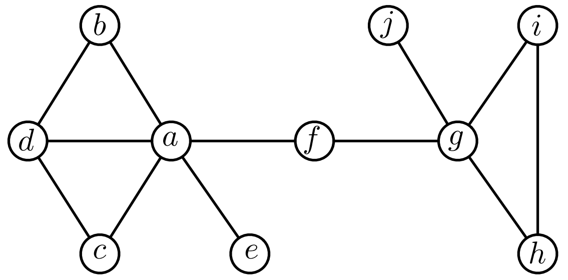


Figure 1. An example graph, to show the different orderings. The DEG order on this graph is $a, g, d, b, c, i, h, f, e, j,$ the DFS and BFS order are coincidentally the same $a, b, d, c, e, f, g, h, i, j,$ the SUB-DEG order is $a, d, b, c, f, g, h, i, e, j$ and the CLUST ordering is $a, b, c, d, e, f, g, h, i, j.$

in the set of vertices already chosen in the order. Or more precisely

$$d_{order}(v) = |\{u \in order | (v, u) \in E\}|$$

For this order, the vertex with the maximal subdegree is chosen from the active set of vertices.

Clustering (CLUST) Another criterion for the choice of a vertex from the active set is the clustering coefficient. This is a measure which quantifies how close the neighborhood of a node is to being a clique. This coefficient for a vertex v_i is computed as:

$$c_i = \frac{|\{e_{jk} : e_{jk} \in E \land v_j, v_k \in N_i\}|}{k_i(k_i - 1)}$$

where N_i is the set of neighbors of v_i and $k_i = |N_i|$. Choosing vertices with a higher clustering coefficient also reduces the search space, because more edges in the neighborhood means more restrictions to the set of candidates (i.e. less candidates) in the assignment phase of the backtracking algorithm.

Figure 1 shows an example graph with the 5 described orders.

As we will see in the empirical evaluation, none of these 5 orders dominates others in the sense that it would consistently result in the smallest search trees. Any of these orders can be the best on some instances and our goal is to find the best one before the actual execution of the algorithm.

III. Search-tree size estimation

In this section we will describe the method for estimating the search tree size and the specific choices we made in order to use it for the subgraph isomorphism search.

A. Knuth's method

Since heuristic sampling is the extension of Knuth's method, we give a quick description of this interesting approach. In a nutshell, the method traverses the tree by choosing a random path until reaching a leaf of the tree.

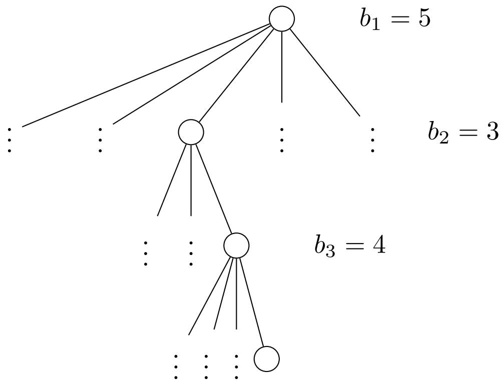


Figure 2. A random walk in the tree, where the branching at each level gives an estimate of the overall tree size. This is the Knuth's estimation method in a nutshell.

This random walk yields one estimation of the tree size as:

$$S_{est} = 1 + b_1 + b_1 b_2 + b_1 b_2 b_3 + \ldots,$$

where b_i is the number of children the node at depth i has (i.e. the branching rate of that node, see Figure 2). By repeating this random walk and averaging the result, the estimation quickly converges very closely to the actual size on many problems.

Our first attempts of using Knuth's method for subgraph isomorphism did not yield satisfactory results. The main reason for this is that the simple sampling assumes a very homogeneous structure of the search-tree. Such structure is present in many classical search problems (such as games and basic combinatorial problems), but it is definitely not present in more complex search problems where the structure varies between branches significantly.

B. Heuristic sampling

Heuristic sampling (HS) addresses the issues of Knuth's method. Instead of following a single path in the tree, HS proceeds on many representative paths at the same time. This approach captures better the heterogeneous nature of search trees.

The central concept of HS is a heuristic function $h : N \to \mathcal{P}$, N being the set of nodes of the search tree and $\mathcal{P}$ being a partially ordered set. This function is called a stratifier and it should be designed in such a way to reflect the main characteristics of the nodes in the search tree. Intuitively, it is a function which maps each node into a value that should describe the shape of the subtree rooted at that node. Two nodes mapped to the same value by the stratifier should have similar (similarly sized) subtrees.

With the stratifier given, the method must now find an estimation for the number of nodes for each $\alpha \in \mathcal{P}$. And from such estimations the entire size of the tree can simply be computed as:

$$S_{est} = \sum_{\alpha} s_{\alpha}.$$

In order for HS to be successful on subgraph isomorphism problem, we had to find a suitable stratifier. One which captures the shape of the tree well enough, but also does not degenerate into a function that has to search the entire tree. We chose to use the function

$$h(n) \to (depth, deg_n),$$

where $depth$ is the depth at which the node is located and deg_n is the number of children the node has. So two nodes will be considered equal if they are at the same depth and they have the same number of children.

The details of the entire procedure are given in Algorithm 2. The method initiates a priority queue, which will serve as the data structure for holding sample nodes of the search tree. The priority of the elements in the queue is $h(n)$, and the dequeuing removes the element with the maximal value of h. At each step of the iteration, one node is retrieved from the queue and all its children are generated. These children (m) are added to the queue if their $h(m)$ is not yet present in the queue otherwise the weight of the element in the queue is increased (we found a new instance for this stratum) and the old node in the queue is replaced by the newly found with a probability $\frac{w}{w_s}$, where w is the weight of the parent node and w_s is the weight of the newly generated node. A detailed theoretical justification of this method can be found in [5].

Algorithm 2: estimateSize(G,H,order) - estimating the size of the search tree. The return value is a list of estimated sizes for each $\alpha \in \mathcal{P}$.

```
Q = [root, 1)];
sol = {};
while $|Q| > 0$ do
    $(n, w) = Q.dequeue$;
    $sol = sol \cup \{(n, w)\}$;
    $Q = Q \setminus (n, w))$;
    for $m \in$ children of $n$ do
        $\alpha_m = h(m)$;
        if $Q$ contains an element $(s, w_s)$ with $h(s) = \alpha_m$ then
            $w_s = w_s + 1$;
            with probability $\frac{w}{w_s}$ do $s = n$;
        end
        else
            $Q = Q \cup (m, 1)$;
        end
    end
end
return sol
```

IV. Empirical evaluation

All the methods were implemented in python, using the igraph [10] library to handle basic graph operations.

For testing the subgraph isomorphism algorithms, one of the most widely used benchmark set is the Amalfi testset [11]. It contains a large amount of different graphs, from Erdös-Renyi random graphs, to highly regular meshes. For this evaluation we chose a subset of 100 instances from this benchmark set. All the chosen graphs are random Erdös-Renyi graphs. The main reason for the choice of such a limited subset is that larger instances are to difficult for such a basic backtracking procedure without more sophisticated pruning techniques. So in order to obtain the exact solutions for all the instances, we had to choose smaller graphs.

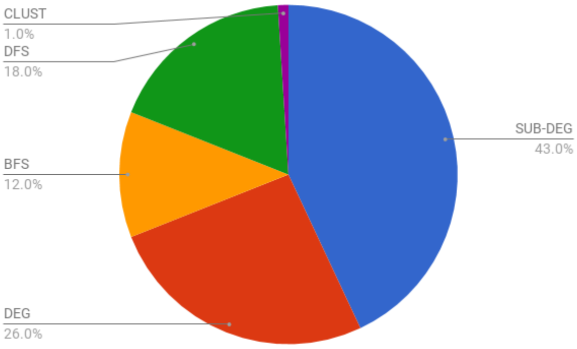


Figure 3. The best orders for each tested instance. Every order is the best on at least one input instance, showing the diversity and importance of predicting the performance for each order.

We first ran the backtracking algorithm on all instances and measured the size of the search tree for each of them. We could also measure the execution time, since it is nicely correlated with the size of the tree. This is simply because the order remains static during the execution and therefore there is no particular overhead for the computation of different orders. But it is better to choose a machine independent measure, which makes the evaluation more robust, so the tree size was the most natural choice. To obtain a more reliable number, each estimation was repeated 100 times, and the average was computed.

First let us demonstrate that these five orders exhibit a large variance in performance on different instances. Figure 3 shows which order was the winner on 100 instances of the benchmark set. We can see that all orders have instance on which they yield the smallest search tree. The clustering order is the winner on only one instance, but the other orders are significantly more successful.

Figure 4 shows also how the sizes of the search trees differ for each instance. The graph shows a scatterplot of relative differences for each instance:

$$\frac{s_{max} - s_{min}}{s_{min}},$$

where s_{max}, s_{min} are the largest and smallest tree sizes between the 5 orders. The y-axis in logarithmic, since in some instances the largest tree size was more than 100 times larger than the smallest one. For the majority of instances the difference is more than 100%, which in practice means that in the worst case, the algorithm would run much longer than in the best case.

Next, we use the heuristic sampling for predicting the tree size for each test instance and each order, altogether

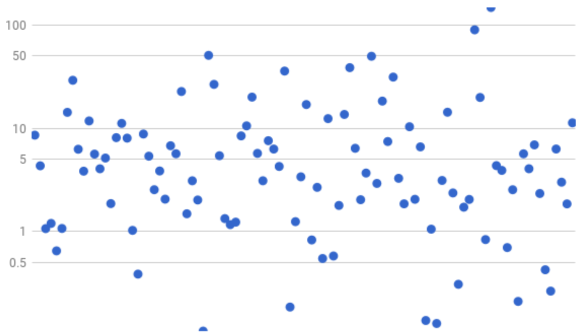


Figure 4. The scatterplot shows the relative difference of the tree size between the best order and worst order on each of the 100 instances. The y-axis is logarithmic.

500 predictions. In order to compare the efficiencies between instances we compute the relative error

$$\frac{|T| - S_{est}}{|T|},$$

where T is the tree size and S_{est} is the estimated size. Figure 5 shows the distribution of these relative errors for the 500 predictions. We can see there are a few predictions that were not that good, but the vast majority of predictions is within 20% of the actual size of the tree.

Since we are investigating if the sampling can be successfully used as a predictor in the backtracking algorithms, the error in the prediction is not the most important factor, What is more important is the ranking accuracy of the method. For each instance we are interested in finding the best possible order in advance. For this reason we now show the distribution of the relative difference between the best tree size, and the tree size of the chosen order (both actual, not the predictions):

$$\frac{|T_{best}| - |T_{chosen}|}{|T_{best}|}.$$

The results for this measurement are shown in Figure 6. As we can see, more than 80% of all chosen orders were either exactly the best order or within 5% of the best order. On one instance the predicted winner performed very badly (more than twice the size of the actual best order), but on the rest of the instances the predictions are very feasible for practical applications.

V. Conclusions and Future Work

There is no silver bullet for solving hard (NP-hard) problems, however many pragmatic approaches achieve remarkable results, making certain problem much more practical than its theoretical properties would suggest. Usually there are many different approaches to solving a certain problem, and many times these approaches vary drastically from instance to instance. We presented a possible solution for further improving an existing set of algorithms by providing a mechanism for choosing the best algorithm for the instance at hand, from a potentially large bag of candidate algorithms.

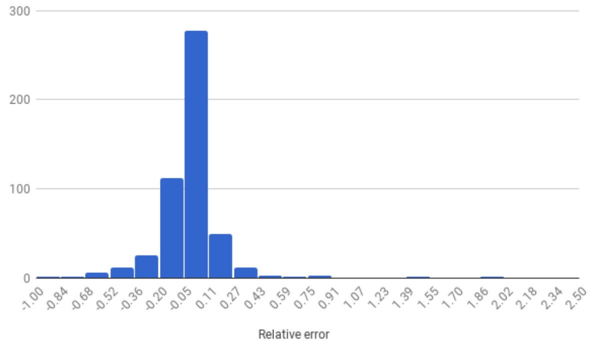


Figure 5. The histogram of relative errors for each prediction (100 instances and 5 orders, i.e. 500 predictions). Two predictions were more than twice the actual number of tree nodes, however most of the predictions were very accurate.

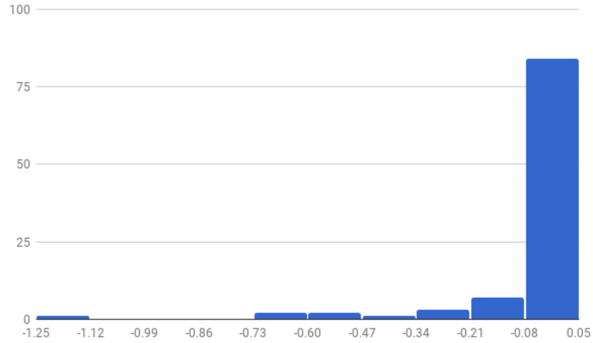


Figure 6. The relative difference between the chosen order and the best order for an instance. In 80 % of cases the chosen order in either the best one or within 5% of the best. In one case the chosen order was twice as bad as the best one.

This is an initial test of the feasibility of the method. In order for this method to become practical we need to involve all the pruning techniques of the more advanced algorithms and see if the sampling is so efficient also in that context. Another research direction is the possibility for new stratifiers in the heuristic sampling. Stratifiers with even more information about the underlying solution could better describe the shape of the tree, resulting in more accurate estimations.

And finally, since the sampling gives us a lot of information about the future performance of the algorithm it could also be used to make the ordering adapt to changes it sees ahead. During the search, the sampling could provide an estimate of the search tree still to be explored. If the size of the tree is too big, the order can change and thus dynamically adapt to different parts of the graph.

References

[1] Dimitris K Agrafiotis, Victor S Lobanov, Maxim Shemanarev, Dmitrii N Rassokhin, Sergei Izrailev, Edward P Jaeger, Simson Alex, and Michael Farnum. Efficient Substructure Searching of Large Chemical Libraries: The ABCD Chemical Cartridge. *J. Chem. Inf. Model.*, 2011.

[2] John M Barnard. Substructure searching methods: Old and new. *J. Chemical Information and Computer Sciences*, 33(4):532–538, 1993.

[3] S. Batra and C. Tyagi. Comparative analysis of relational and graph databases. *Int'l J. Soft Computing & Engineering*, 2012.

[4] Vincenzo Bonnici and Rosalba Giugno. On the variable ordering in subgraph isomorphism algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 14(1):193–203, January 2017.

[5] Pang C Chen. Heuristic sampling: A method for predicting the performance of tree searching programs. *SIAM Journal on Computing*, 21(2):295–315, 1992.

[6] Uroš Čibej and Jurij Mihelič. Improvements to ullmann's algorithm for the subgraph isomorphism problem. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(07):1550025, 2015.

[7] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recongnition. *Int'l J. Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[8] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd annual ACM symposium on Theory of computing - STOC '71*, pages 151–158. ACM Press, 1971.

[9] Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1367–1372, 2004.

[10] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[11] M. De Santo, P. Foggia, C. Sansone, and M. Vento. A large database of graphs and its use for benchmarking graph isomorphism algorithms. *Pattern Recognition Letters*, 24(8):1067–1079, May 2003.

[12] Wenfei Fan. Graph pattern matching revised for social network analysis. In *Proc. 15th International Conference on Database Theory - ICDT '12*, page 8. ACM Press, March 2012.

[13] Donald E. Knuth. Estimating the efficiency of backtrack programs. Technical report, Stanford, CA, USA, 1974.

[14] Jianzhuang Liu and Yong Tsui Lee. Graph-based method for face identification from a single 2D line drawing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(10):1106–1119, 2001.

[15] Christine Solnon. AllDifferent-based filtering for subgraph isomorphism. *Artificial Intelligence*, 174(12-13):850–864, August 2010.

[16] Julian R Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

[17] Julian R Ullmann. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *Journal of Experimental Algorithmics (JEA)*, 15:1–6, 2010.

[18] Uroš Čibej and Jurij Mihelič. Search strategies for subgraph isomorphism algorithms. In *International Conference on Applied Algorithms, ICAA*, pages 77–88, 2014.

Overview of the Handwriting Processing for Clinical Decision Support System

Zuzana Dankovičová
Technical University of Košice
Department of Computer and Informatics
Slovak Republic
Email: zuzana.dankovicova@tuke.sk

Peter Drotár
Technical University of Košice
Department of Computer and Informatics
Slovak Republic
Email: peter.drotar@tuke.sk

Juraj Gazda
Technical University of Košice
Department of Computer and Informatics
Slovak Republic
Email:juraj.gazda@tuke.sk

Liberios Vokorokos
Technical University of Košice
Department of Computer and Informatics
Slovak Republic
Email: liberios.vokorokos@tuke.sk

***Abstract*—A clinical decision support system (CDSS) is an information system used in healthcare, designed to help with the clinical decision. It supports medical diagnostics, treatment and prevention. CDSS systems exist in many forms and make use of the different types of data. In this paper, we focus on CDSS based on handwriting and provide the overview of current state-of-the-art in the handwriting processing for CDSS with the focus on dysgraphia. The recent advances in technology facilitate possibility to analyse the handwriting not only as image but accurately analyse components of handwriting such as movement in horizontal and vertical direction, pressure and movement above the surface. We outline tasks used for the handwriting acquisition, handwriting features, classification models and latest achieved results in area of handwriting processing. We identify currently the the most useful approaches for dysgraphia rating and classification. Our ultimate goal is to design a CDSS to make the diagnostics and monitoring of the impaired handwriting objective more effective.**

Keywords: dysgraphia, handwriting, decision support system.

I. Introduction

The recent advances in the technology and reduced manufacturing cost of the electronic devices allowed for widespread usage of high tech devices in many areas of medicine. The vision of modern clinicians laid the foundation for the emergence of artificial intelligence in medicine (AIM). The formal definition of AIM is as follows. *"For the system of artificial intelligence in medicine, respectively medical practice, is considered artificial intelligence which is able to diagnose and prescribe treatment for the patient. Unlike medical applications that are based on statistical and probabilistic programming techniques, the AIM is based on symbolic models representing disease and its relationship to patients."* [1]. With the increased usage the term gained more meanings and was later replaced by name clinical decision support system (CDSS).

According to the study of Menekse et al. [2], CDSS improves the success rates of diagnosing and treatment phases. CDSS can significantly improve healthcare by using all available medical knowledge to offer reasonable person-specific recommendations during the treatment process [3]. There are many studies that have been successfully applied in different clinical fields. There are various applications of the CDSS in cardiology [4], brain diseases such as Alzheimer diseases [5] and Parkinson diseases [6], and speech pathologies [7]. The results indicate that applying CDSS improves the quality of diagnosis and treatment processes and CDSS can be used as an additional education aid for novice medical staff.

CDSS can be build upon direct processing of different biomedical signals such as ECG, EEG, etc. or alternatively CDSS can make use of existing health records to support the clinician decision. In this paper, we focus on the first category, i.e. CDSS that evaluate data collected from patient and provide diagnostic recommendation. Particularly, we focus on CDSS for handwriting processing, since acquisition of handwriting is non-invasive, low-cost and reflect the syndromes of many frequent disorders. More precisely we focus on dysgraphia and the possibilities of processing and monitoring of the impaired handwriting for CDSS.

Despite the widespread use of computers, handwriting remains an important skill that deserves greater attention from educators and health practitioners [8]. Children spend 31% to 60% of their school day performing handwriting and other fine motor tasks [9][10]. Difficulty in handwriting interfere with the academic achievement. Handwriting is both a tool for communication and a necessary life skill. It is still the most immediate form of graphic communication [11].

The consequences of poor handwriting for academic performance have been well documented [10]. Researchers suggested that handwriting difficulties might have serious consequences for the students overall academic success, emotional well-being, behavior and attitude [8][12]. These findings reinforce the importance of identifying handwriting difficulties as soon as possible, both as a preventive and as a corrective aid

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

[13]. Sandler et al. [14] found that children with writing disorders had a tendency towards lower mathematics achievement, low verbal IQ, and increased attentional difficulties.The reason underlying a child's writing difficulty may often be unclear as elements of the writing process are closely interwoven. For example, difficulty in spelling can contribute to difficulty in expression of ideas or note-taking [15]. Handwriting is often judged and seen as a reflection of an individuals intelligence or capabilities. It is illustrated by several studies in which lower marks were consistently assigned to students with poor handwriting and higher marks given to those with legible handwriting despite similar content [16].

In [17], Prattichizzo et al. focus on the analysis of human hand movements during handwriting tasks, where investigated and compared the digital handwriting task with the index fingertip and with a stylus. They were inspired, for example, by Hermsdörfer et al. [18], who exploited information on the forces applied by the fingers on a pen, and the relative grip kinetics, to inform diagnosis and treatment of handwriting disorders. In [19] a device called the Kinetic Pen was presented and Shim et al. [20] adopted this device to investigate synergistic actions of hand-pen contact forces during circle drawing tasks. They were inspired also by Kushki et al. [21], who analyzed how some handwriting kinetic and kinematic parameters vary during a prolonged writing task in children with and without handwriting deficiency. This motivates Prattichizzo et al. to applying manipulability analysis to the human hand. Their results confirm that using a stylus is in general more convenient for performing precision tasks, such as drawing, writing, signing a document, etc [17]. A fingertip interaction is more suitable in tasks with a wide workspace and where a lower precision is required, for example selecting icons, answering a call, etc. They also developed mathematical tools that allow the analysis of other aspects of digital handwriting, such as the dependency of writing performance on hand biomechanical properties, or on hand postures [17].

In the following sections, we will present an overview of the techniques in handwriting processing and consider their application in CDSS.

II. CDSS overview

CDSS are used to process, analyze, and retrieve information from available resources needed by physician to decide in the particular situation. Their goal is not to replace the physician, but to support him by streamlining the decision-making processes, preventing the mistakes in human decision-making and providing early warnings [22].

The physician interacts with CDSS, which analyze patient data and his anamnesis. This help to predict potential critical events from drug intolerance, drug interactions or the signs of the disease. The physician uses both the own knowledge and the knowledge of CDSS to better analyze the patients condition. For example, CDSS suggests, the diagnosis or treatment process. Based on this, the physician selects useful information and discards erroneous CDSS designs.

A. Classification of CDSS

DDSS (Diagnosis Decision Support System) is a specific typ of CDSS. It suggests a set of appropriate diagnoses, based on patien data. The physician then takes the output from DDSS and determines, which diagnoses are relevant and which are not [23]. Another type of CDSS is CRB (Case-based reasoning) system. This system is used in radiotherapy for brain cancer patients. Thy psysician review the recommended treatment plan and determine its effectiveness [24].

There are two main types of CDSS:

- *Knowledge-based* - it has three parts: the knowledge base, an inference engine, and a mechanism to communicate. The knowledge base contains the rules and data. Data are most often in the form IF-THEN. For example, if a patient takes medication, the system creates a rule:
  "IF drug X is taken and patien starts to use another drug Y, THEN alert user."
  The inference engine combines the rules from the knowledge base with the patients data. The communication mechanism allows the system to show the results to the user [23].
- *Non-knowledge-based* - It uses a form of artificial intelligence called machine learning [25]. Based on previous experience, the system is able to learn to understand, for example, the symptoms of identifying various diseases. It can also find patterns in clinical data and suggest procedures for physicians. They can be useful as post-diagnostic systems. They are suitable for the development of clinical guidelines. Non-knowledge-based systems are neural networks, Petri networks and agent systems.

Timing of the system use is either a pre-diagnosis, diagnosis, or post-diagnosis. Pre-diagnostic systems help the physician to prepare the diagnosis. The systems used during diagnostics check the pre-decision of the physician in order to improve their final results. Post-diagnostic systems serve to collect patient data and their anamnesis, to predict future events.

III. Case studies of handwriting processing analysis

In this section, we focus on the case studies of the pathological handwriting analysis, that earned considerable interest in the field. In particular, we focus on three studies:

1) Identification and rating of developmental dysgraphia by handwriting analysis [26],
2) Identifying developmental dysgraphia characteristics utilizing handwriting classification methods [27],
3) Lifts and stops in proficient and dysgraphic handwriting [28].

These studies address various components of understanding the pathological handwriting, such as assessing intelligibility of handwriting or its impaired, identifying the patterns of dysgraphia in handwriting, and offer solutions of pathological handwriting conditions.

A. Identification and rating of developmental dysgraphia by handwriting analysis

Amongst the studies in handwriting pathology, work of Mekyska et al. [26] provides an innovative solutions from several points of view. The objective of this study was to propose a method that can be used for automated diagnosis of difficulty in the production of written language. The process of adopting the letters form begins at the first grade, sometimes even earlier. Around the age of eight unification occurs for an automatic proficient manner of letters production [29]. Children at this age who do not succeed in developing the proficient handwriting face developmental dysgraphia [21][30]. All participants on this study were born in Israel and used the Hebrew language. Two groups of handwriters (proficient and dysgraphic), each consisting of 27 third-grade male and female pupils, aged 8 and 9, were included in the study.

Two short and practical questionnaires enable dysgraphia identification. One is designated for teachers or parents (HPSQ - Handwriting Proficiency Screening Questionnaire) and one for the child's self-report (HPSQ-C - Handwriting Proficiency Screening Questionnaire for Child). The HPSQ [31] is a ten-item questionnaire developed to identify school-aged children with handwriting difficulties based on their teacher's observation. The ten items cover the most important indicators of handwriting deficiencies following the definition of dysgraphia offered by Hamstra-Bletz and Blote [32], including legibility, performance time, and physical and emotional well-being. These were scored on a five-point Likert scale (from 0 - never to 4 - always), and a final score is summed. In the case of testing, are know HPSQ score given by clinician and the HPSQ score estimated by the system.

To suppress the individuality in handwriting and to emphasize dysgraphic features, simple intrawriter normalization method based on subtraction have been introduced [26]. They proved increase discrimination accuracy by 4% and decrease HPSQ score estimation error by 3.48%. In their work, they employ complex parametrization based on ten kinematic measures, 34 nonlinear dynamic, and other seven features. They considered a large and complex set of the features that quantify handwriting geometry, dynamics, tremor, pressure, and altitude. To analyze the whole handwriting and particular strokes as well, they used kinematic and spatio-temporal features as speed, velocity, acceleration, jerk, normalized jerk, width, height, orientation, duration, and length [33][34][35][36].

They proposed system of automatic dysgraphia discrimination with 96% sensitivity and specificity of methods designated for handwriting deficiency evaluation. They found that coefficient of variation of altitude has one of the best discrimination power among features. This means that children with dysgraphia are not able to hold pen in a stable position, and therefore, the pen tilt is changing a lot in time. A system is able to automatically rate development dysgraphia with approximately 10% HPSQ estimation error.

B. Identifying developmental dysgraphia characteristics utilizing handwriting classification methods

In [27], Rosenblum and Dror used machine learning methodologies to infer a statistical model, which is capable of discriminating dysgraphic from proficient handwriting with approximately 90% accuracy. Since the model provides 90% sensitivity and a specificity of 90%, it is the first step toward future use as an effective standard indicator for dysgraphia detection. The aim of this study was to develop a statistical model for discriminating between dysgraphic and proficient handwriting, based on their performance characteristics. The most robust way to build such models is by use of the machine learning techniques, which have shown great effectiveness in finding discriminative patterns in the data [37].

The 99 third graders aged eight and nine years old (proficient and dysgraphic) were selected to participate in this study. Participants were from northern Israel and also used the Hebrew language as in [26]. They were identified as having/not having handwriting difficulties/ dysgraphia by means of the standardized HPSQ [31].

The work contained four tasks, but, only three were used. Task A was repeating of a six-word sentence from a well-known children's song with features as total task time, total task time in fifth quintile, total task time in first quintile, on-paper fraction, lamed height ratio, mean distance between letters, pressure mean, pressure standard deviation, transpositions and deletions. Task B has not been used. Task C included a sequence of repeating a single letter with features as number of segments, number of primitive shapes and total task time. Task D consists of identical well-separated loops, drawn with a single stroke, where features are number of loops, number of crossing loops, number of contained loops, standard deviation of loop heights, the mean width (horizontal) of loops, total task time and negative curvature fraction.

In the study they used ComPET, a standardized and validated handwriting assessment utilizing a digitizing tablet and online data collection and analysis software [38].

C. Lifts and stops in proficient and dysgraphic handwriting

According to several studies [39][40][41], dysgraphic children showed a lack of continuity and fluency in handwriting, significantly longer lift duration of the pen or extended stops intervals between strokes. As a matter of fact, stops in handwriting may be more effective than the lifts in revealing difficulties with writing. Stops occurring while the pen stays in contact with the paper may reflect difficulties in the mechanics of writing, such as those described in the case of dysgraphic children [32].

In [28], they compared 26 proficient adults, 39 proficient children and 16 dysgraphic children in the same writing task. All participants were from Marseille, used the Roman writing system and all children were attending third grade. The participants wrote with an ink pen on a sheet of paper fixed to a digitizer tablet. They only wrote the same word 'lapin' (rabbit) eight times at normal handwriting speed, and eight times at a fast speed. At fast speed, participants were required to write as fast as possible while maintaining legibility.

The aim of this study was to focus exclusively on low-level graphomotor processes. Five spatiotemporal variables were used as dependent variables: mean writing velocity, number of lifts, mean duration of lifts, number of stops and mean duration of stops. In study, a distinction was made between short, normal, and long abnormal stops. Paz-Villagrán et al. in [28] assumed that stops shorter or equal to 35 ms correspond to the normal stops of handwriting. Longer abnormal stops (longer than 35 ms) are particularly interesting because they may result from a dysfluency problem related to neuromotor noise in children with dysgraphia [42] and have a longer duration than the normal stops.

Adults had less lifts than children, and only dysgraphic children exhibited a significant difference. Proficient children tend to stop more often than adults because they had a tendency to write in a discrete way. This phenomenon was still more marked in dysgraphic children who often had longer stops and, moreover, stopped within strokes. The number and duration of stops decrease when dysgraphic children write faster, so they are more fluent at faster speed and the increase of velocity did not induce a decrease in legibility.

The results of this study suggest that the pen stops are more appropriate than pen lifts in differentiating the handwriting fluency for identifying dysgraphia.

IV. Conclusions

There has been considerable interest to develop an automatic assessment system of handwriting for dysgraphia. Such a system should provide more accurate, objective solution in order to support handwriting therapy. We reviewed and compared several recently published studies for the handwriting processing that represent cornerstone for the future decision support systems.

The review has shown that there is huge space for further development. The state-of-art features are usually limited to basic temporal and kinematic features and are far beyond the developments in the area of e.g. speech processing. Introduction of the new features based on spectral analysis, signal decomposition, non-linear analysis, etc. would provide deeper insight on the characteristics of the handwriting. This may be of significant importance in assessment of the handwriting.

Current studies focus almost solely on identifying dysgraphia among healthy subjects. However, experienced therapist is capable of identifying dysgraphia within few minutes. Ultimate goal in this case is to rate the level of handwriting impairment so it can be tracked during the therapy.

One significant aspect is the design of the handwriting tasks. The literature review have shown that not all handwriting tasks reflect the handwriting deterioration. The selection of the template for the data acquisition should be preceded with the careful review and design.

Another aspect that needs to be considered is the user interface. If the decision support systems should be used in everyday practice it has to be designed with the focus on high usability. In general, the main reason for the failure of CDSS is they low usability. This is even more important in the case of dysgraphia, where the target group children with special needs such as children with developmental disabilities.

Even though we focus on dysgraphia, the CDSS once designed, can be with some modifications, used also for other diseases.

Acknowledgment

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-15-0055 and contract No. APVV-16-0211

References

[1] E. Coiera, "Guide to the Health Informatics 2nd edition," 1997-2003, ISBN 0340 764252.

[2] G. G. Menekse, N. E. Cagiltay and G. Tokdemir, "Patient safety & clinical decision support systems (CDSS): A case study in Turkey," E-Health and Bioengineering Conference (EHB), Iasi, 2015, pp. 1-4, 2015.

[3] S. Y. Ji, K. Najarian, R. Smith and T. Huynh, "Computer Aided Traumatic Pelvic Injury Decision Making," Proceedings of the Southern Association for Information Systems Conference, Richmond, VA, USA (2008).

[4] I. Kadi, A. Idri and S. Ouhbi, "Quality evaluation of cardiac decision support systems using ISO 25010 standard," 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, 2016, pp. 1-8.

[5] C. M. Carvalho, D. Christina, M. Saade, A. Conci, F. L. Seixas and J. Laks, "A clinical decision support system for aiding diagnosis of Alzheimer's disease and related disorders in mobile devices," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

[6] M. A. Raza, Q. Chaudry, S. M. T. Zaidi and M. B. Khan, "Clinical decision support system for Parkinson's disease and related movement disorders," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 1108-1112.

[7] R. Gupta, T. Chaspari, J. Kim, N. Kumar, D. Bone and S. Narayanan, "Pathological speech processing: State-of-the-art, current challenges, and future directions," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 6470-6474.

[8] K. P. Feder and A. Majnemer, "Handwriting development, competency, and intervention," Develop. Med. Child Neurol., vol. 49, no. 4, pp. 312–317, 2007.

[9] K. McHale and S. A. Cermak,"Fine motor activities in elementary school: preliminary findings and provisional implications for children with fine motor problems," Am J Occup Ther 46: 898–903, 1992.

[10] A. Kushki, H. Schwellnus, F. Ilyas and T. Chau, "Changes in kinetics and kinematics of handwriting during a prolonged writing task in children with and without dysgraphia," Res. Develop. Disabilities, vol. 32, no. 3, pp. 1058–1064, 2011.

[11] R. Sassoon, "Handwriting: A New Perspective. Cheltenham," UK: Stanley Thornes, 1990.

[12] S. T. Peverly, P. C. Vekaria, L. A. Reddington, J. F. Sumowski, K. R. Johnson and C. M. Ramsay, "The relationship of handwriting speed, working memory, language comprehension and outlines to lecture notetaking and test-taking among college students," Appl. Cognitive Psychol., vol. 27, no. 1, pp. 115–126, 2013.

[13] M. R. I. Martins, J. A. Bastos, A. T. Cecato, M. de Lourdes Souza Araujo, R. R. Magro and V. Alaminos, "Screening for motor dysgraphia in public schools," J. Pediatria, vol. 89, no. 1, pp. 70–74, 2013.

[14] A. D. Sandler, T. E. Watson, M. Footo, M. D. Levine, W. L. Coleman and S. R. Hooper, "Neurodevelopmental study of writing disorders in middle childhoo," Dev Behav Pediatr 13: 17–23, 1992.

[15] N. Mather and R. Roberts, "Informal Assessment and Instruction in Written Language: A Practitioner's Guide for Students with Learning Disabilities. Brandon," VT: Clinical Psychology Publishing Company, 1995.

[16] V. Connelly, S. Campbell, M. MacLean and J. Barnes,"Contribution of lower order skills to the written composition of college students with and without dyslexia," Dev Neuropsychol 29: 175–196, 2006.

[17] D. Prattichizzo, L. Meli and M. Malvezzi, "Digital Handwriting with a Finger or a Stylus: A Biomechanical Comparison," in IEEE Transactions on Haptics, vol. 8, no. 4, pp. 356-370, Oct.-Dec. 1 2015.

[18] J. Hermsdörfer, C. Marquardt, A. S. Schneider, W. Fürholzer and B. Baur, "Significance of finger forces and kinematics during handwriting in writer's cramp," Human movement science, vol. 30, no. 4, pp. 807–817, 2011.

[19] A. W. Hooke, J. Park and J. K. Shim, "The forces behind the words: Development of the kinetic pen," Journal of biomechanics, vol. 41, no. 9, pp. 2060–2064, 2008.

[20] J. K. Shim, A. W. Hooke, Y. S. Kim, J. Park, S. Karol and Y. H. Kim, "Handwriting: Hand–pen contact force synergies in circle drawing tasks," Journal of biomechanics, vol. 43, no. 12, pp. 2249– 2253, 2010.

[21] A. Kushki, H. Schwellnus, F. Ilyas and T. Chau, "Changes in kinetics and kinematics of handwriting during a prolonged writing task in children with and without dysgraphia," Research in Developmental Disabilities, vol. 32, no. 3, pp. 1058–1064, 2011.

[22] K. Kawamoto, C. A. Houlihan, A. E. Balas and D. F. Lobach, "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success," Bmj 330.7494, 2005.

[23] E. S. Berner, "Clinical Decision Support Systems," New York, NY: Springer, 2007.

[24] G. Khussainova, S. Petrovic and R. Jagannathan, "Retrieval with clustering in a case-based reasoning system for radiotherapy treatment planning," Journal of Physics: Conference Series, 616 (1): 012013, ISSN 1742-6596, 2015.

[25] T. Syeda-Mahmood, "Plenary talk: The Role of Machine Learning in Clinical Decision Support," SPIE Newsroom, March 2015.

[26] J. Mekyska, M. Faundez-Zanuy, Z. Mzourek, Z. Galaz, Z. Smekal and S. Rosenblum, "Identification and Rating of Developmental Dysgraphia by Handwriting Analysis," in IEEE Transactions on Human-Machine Systems, vol. 47, no. 2, pp. 235-248, April 2017.

[27] S. Rosenblum and G. Dror, "Identifying Developmental Dysgraphia Characteristics Utilizing Handwriting Classification Methods," in IEEE Transactions on Human-Machine Systems, vol. 47, no. 2, pp. 293-298, April 2017.

[28] V. Paz-Villagrán, J. Danna and J. L. Velay, "Lifts and stops in proficient and dysgraphic handwriting, In Human Movement Science," Volume 33, 2014, Pages 381-394, ISSN 0167-9457.

[29] S. Rosenblum and L. Gafni-Lachter, "Handwriting proficiency screening questionnaire for children (HPSQ-C): Development, reliability, and validity," Amer. J. Occupat. Therapy, vol. 69, no. 3, pp. 6903220030, 2015.

[30] A. O'Hare, "Hands up for handwriting," Develop. Med. Child Neurol., vol. 46, no. 10, pp. 651–651, 2004.

[31] S. Rosenblum, "Development, reliability, and validity of the handwriting proficiency screening questionnaire (HPSQ)," Amer. J. Occup. Therapy, vol. 62, no. 3, pp. 298–307, 2008

[32] L. Hamstra-Bletz and A. Blote, "A longitudinal study on dysgraphic handwriting in primary school," J. Learn. Disabilities, vol. 26, no. 10, pp. 689–699, 2003.

[33] G. Luria, A. Kahana and S. Rosenblum, "Detection of deception via handwriting behaviors using a computerized tool: Toward an evaluation of malingering," Cognitive Comput., vol. 6, no. 4, pp. 849–855, 2014.

[34] P. Drotar, J. Mekyska, I. Rektorova, L. Masarova, Z. Smekal and M. Faundez-Zanuy, "A newmodality for quantitative evaluation of Parkinson's disease: In-air movement," in Proc. IEEE 13th Int. Conf. Bioinformat. Bioeng, 2013, pp. 1–4.

[35] P. Drotar, J. Mekyska, Z. Smekal, I. Rektorova, L. Masarova and M. Faundez-Zanuy, "Prediction potential of different handwriting tasks for diagnosis of Parkinson's," in Proc. E-Health Bioeng. Conf., 2013, pp. 1–4.

[36] J. H. Yan, S. Rountree, P. Massman, R. S. Doody and H. Li, "Alzheimer's disease and mild cognitive impairment deteriorate fine movement control," J. Psychiatric Res., vol. 42, no. 14, pp. 1203–1212, 2008.

[37] C. Bishop, "Pattern Recognition and Machine Learning," New York, NY, USA: Springer, 2006.

[38] S. Rosenblum, S. Parush and P. L.Weiss, "Computerized temporal handwriting characteristics of proficient and non-proficient handwriters," Amer. J. Occup. Therapy, vol. 57, no. 2, pp. 129–38, 2003.

[39] S. Rosenblum, P. Weiss and S. Parush, "Temporal measures of poor and proficient handwriters," In R. G. J. Meulenbroek and B. Steenbergen (Eds.), Proceedings of the 10th Biennal Conference of the International Graphonomics Society (pp. 119–125). Nijmegen, The Netherlands: IGS Pub, 2001.

[40] S. Rosenblum, P. L. Weiss and S. Parush, "Product and process evaluation of handwriting difficulties," Educational Psychology Review, 15, 41–81, 2003a.

[41] S. Rosenblum, S. Parush and P. L. Weiss, "Computerized temporal handwriting characteristics of proficient and poor handwriters," American Journal of Occupational Therapy, 57, 129–138, 2003b.

[42] G. P. Van Galen, S. J. Portier, B. C. M. Smits-Engelsman and L. R. B. Schomaker, "Neuromotor noise and poor handwriting in children," Acta Psychologica, 82, 161–178, 1993.

Simulation Study of Critically Loaded Arrays of Pillars

Tomasz Derda
Institute of Mathematics
Czestochowa University of Technology
Czestochowa, Poland
tomasz.derda@im.pcz.pl

Zbigniew Domański
Institute of Mathematics
Czestochowa University of Technology
Czestochowa, Poland
zbigniew.domanski@im.pcz.pl

***Abstract*—A failure in an overloaded system is a complex engineering problem. Arrays of pillars belong to a group of modern nanodevices composed of a large number of identical parts that function as a unit. This is because the pillars are fixed on a flat support and interactions among them emerge due to the support's rigidity. When a growing load is applied to the pillars it induces a sequence of failures among the pillars, decreases the device performance and eventually triggers a catastrophic avalanche of failures. A key aspect of how does such a critical destruction evolve is the so-called load transfer rule from destroyed pillars to the intact ones, where a particular transfer rule reflects a distance on which the pillars interact effectively. A common approach to study arrays-of-nanopillars-crushing processes is to employ an appropriate load transfer rule within computer simulations. In our simulations we use different load transfer rules to analyze amount of allocated memory (m) and CPU time (t) consumed to handle these rules for arrays with an increasing number of pillars (N). Specifically, we discuss distributions of m and t, referred to the employed rules as well as the corresponding mean values: $\overline{m} \sim N$ and $\bar{t} \sim N^{b+c\ln N}$.**

***Index Terms*—computer simulations, memory and time complexity, multicomponent system**

I. Introduction

Computer simulations involving algorithms enable scientists and engineers to connect theoretical concepts and their prospective applications in an efficient way. An efficiency of a given algorithm depends on many factors but from the hardware perspective its processing speed and the amount of allocated memory are the most important. Only these two factors show that a user who is looking for a tool to solve a computationally demanding problem has to carefully choose among accessible algorithms. It is also the case in the context of studies involving arrays of nanopillars.

Arrays of nanopillars belong to a wide class of multicomponent modern nanodevices which are encountered in a variety of applications. They are composed of a large number of almost identical parts that function as a unit. A possible sequence of failures among these components decreases the device performance and may eventually lead to a catastrophic avalanche of failures. This is because, once the system is subjected to an increasing load it begins to fail immediately when the internal load intensity equals or exceeds the critical value of weakest components and the failure develops in a form of avalanches of simultaneously damaged elements. More specifically, avalanches appear when an increasing load eliminates an element from the working community in such a way that this exclusion alters the internal load pattern sufficiently to trigger the failure of the other elements and, in consequence, provoking a wave of destruction. A common approach to study avalanches of failures is to apply different statistical models. Among them, the Fibre Bundle Models (FBM) and Random Fuse Models [1]–[4] are frequently employed in problems related to technological applications. Our system is a grid of components represented by a collection of pillars located at nodes of a square lattice, see Fig. 1, and then analyzed within a Fibre Bundle Model framework [5]–[7].

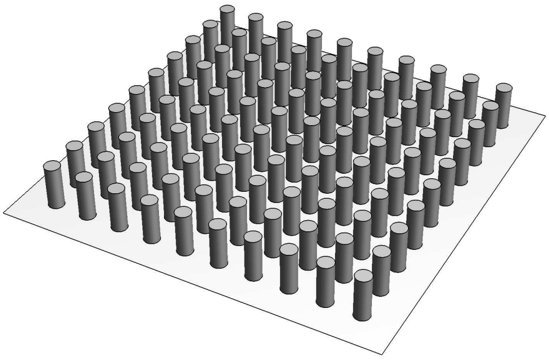


Fig. 1. Schematic view of an array of nanopillars.

We restrict our analysis to the case where each component is characterized by two states: working or failed. We also assume that failed components are not repairable. In our simulations, an ensemble of N components is subjected to a growing load F, that systematically eliminates weak components and involves avalanches of failures. This means that when a component breaks, its load is transferred to the other intact elements and thus the probability of subsequent failures increases. It is assumed that the pillars are of equal height and cross sectional areas, and whose strength-thresholds σ_{th}, $i = 1, 2, ..., N$ to an applied axial load are independent random variables. Pillar-strength-thresholds are drawn from the Weibull distribution with cumulative distribution function

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

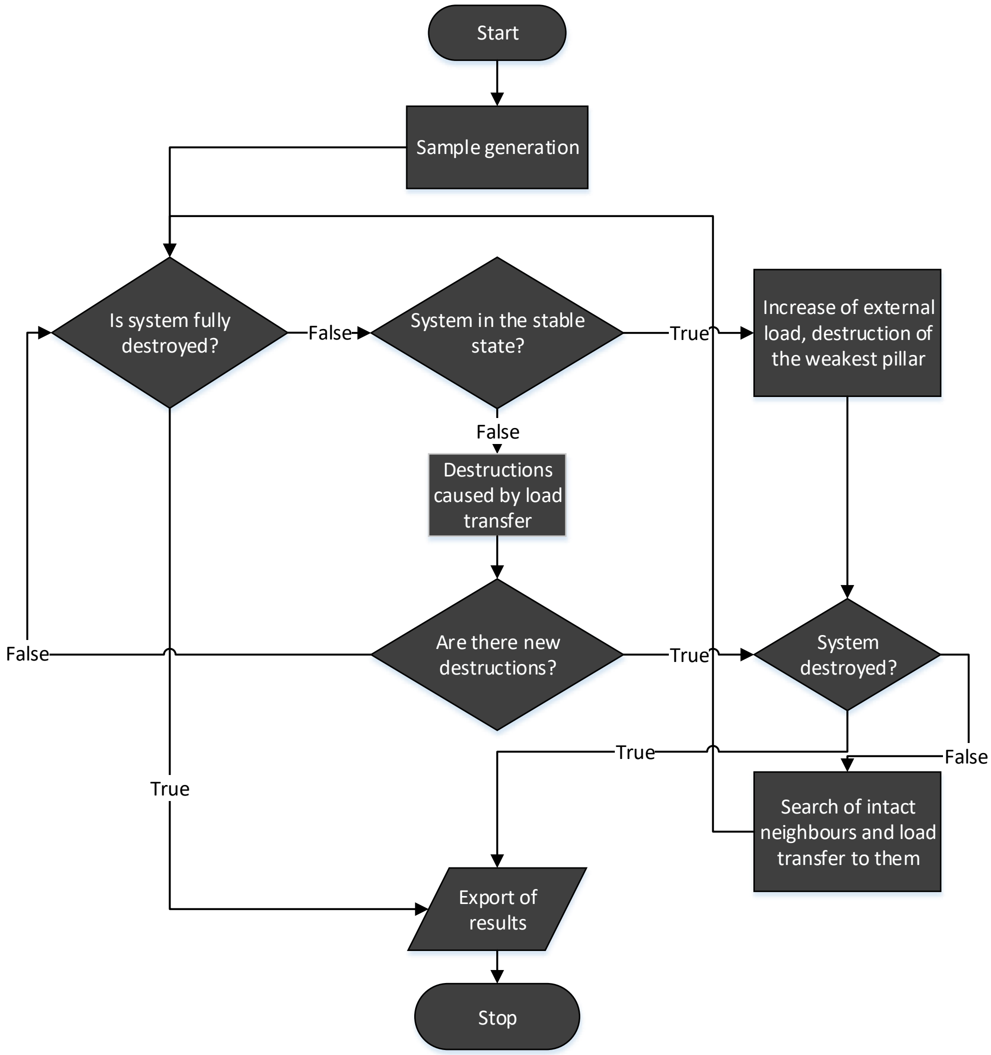

Start
Sample generation
Is system fully destroyed?
False
System in the stable state?
True
Increase of external load, destruction of the weakest pillar
False
Destructions caused by load transfer
Are there new destructions?
False
True
System destroyed?
True
False
True
Search of intact neighbours and load transfer to them
Export of results
Stop

Fig. 2. Flowchart of the LLS algorithm.

[10], [11]

$$P(\sigma_{th}) = 1 - \exp\left[-\left(\frac{\sigma_{th}}{\sigma_0}\right)^{\rho}\right] \tag{1}$$

where σ_0 and ρ are Weibull's scale and shape parameters, respectively. In this work, $\rho = 2$ and $\sigma_0 = 1$.

Damage evolution in progressively loaded physical systems is a highly complex phenomenon that can not be treated analytically. Apart from experimental works computer simulations are the only reliable source of information that can be used to model the evolving damage. Our paper is motivated by an absence of estimations of how the CPU time and memory scale with the number of systems' components in simulations of the FBM. According to our knowledge only a few published works analyse complexity of programs used in simulation of failures in systems with load transfer. Examples include such papers as [8], [9]. However, they are related to one dimensional systems.

II. Computation method

During the loading process, sequences of simultaneous ruptures of several components take place. Within a short interval between consecutive failures the load carried by the destroyed component is transferred only to some of intact, i.e. still working components. In order to handle the load partitioning into groups of working components we employ different load transfer rules. Among them, a special attention we pay to a so-called local load sharing (LLS) rule. Within the LLS a load carried by the destroyed component is transferred only to its closest intact neighbours and this means that numbers of nearest intact neighbours vary during the loading process. Because of such a limited-range-load-transfer, the distribution of intrinsic load is not homogeneous giving rise to appearance of regions of load accumulation throughout the entire system. Therefore, the LLS rule represents short range interactions among pillars. This is in contrast to a global load sharing (GLS), for which long range interactions are assumed. In the GLS scheme the load from the broken pillar is equally redistributed on all intact pillars in the system. The last analysed rule is based on Voronoi polygons - the extra load is equally redistributed among the elements lying inside the Voronoi regions generated by a group of elements destroyed within an interval of time taken to be the time step. This rule, called Voronoi load sharing (VLS), merges the GLS and LLS rules [7].

In this work we focus on the algorithm with the LLS rule. However, results referred to other load transfer rules are also presented and discussed. There are at least two reasons why we concentrate on the LLS rule: (i) using this rule the algorithm is less time consuming comparing to the case when the VLS rule is applied and (ii) as opposed to the GLS transfer rule the damage evolution under the LLS rule can not be treated analytically.

Each our numerical experiment starts from loading all components by the same load, equal to the minimal failure threshold. As a consequence the weakest pillar crushes and its load is transferred to other intact pillars according to assumed load transfer rule. The increasing load imposed on the intact components leads to other failures, after which each intact component bears growing load. If the load transfer does not trigger further failures, a stable configuration emerges meaning that this present value of F is not sufficient to provoke destruction of the entire system, and its value should be increased. In the simulations we applied a quasi-static loading procedure, i.e. if the system is in a stable state the external load increases uniformly on all the intact components by an amount δF sufficient to break the weakest-working component and then the increase of load stops until a new stable state emerges. The procedure is repeated until complete destruction of the system.

Above described loading process serves to built an algorithm to simulate destruction processes involving all load transfer rules. Fig. 2 graphically represents our algorithm in a case when the LLS rule is applied.

III. Results and Discussion

Our program code for the simulation of damage evolution in nanopillar arrays was developed in Mathematica v. 11 (Wolfram Inc.) and executed with the help of Dell Precision T3420 workstation with the Intel Core i7-6700 processor and 16 GB of RAM. Below we list three Mathematica built-in functions whose values have been collected from simulations:

- MaxMemoryUsed[*expr*] – the function gives the maximum number of bytes used during the evaluation of *expr*.
- Timing[*expr*] – the function enables to measure CPU time spent in the Wolfram Language Kernel during the evaluation of *expr*.
- AbsoluteTiming[*expr*] – the function enables to measure the absolute number of seconds in real time (i.e. wall-clock time) that have elapsed during the evaluation of *expr*.

We have recorded values of these functions for each single simulation. Time-intensive computations have been conducted for different system sizes. In order to get reliable statistics, each simulation was repeated at least 5×10^3 times.

TABLE I
MaxMemoryUsed

Transfer rule	Coefficients	
	a	b
GLS	31.9113	6539.77
LLS	442.534	−10921.50
VLS	517.056	−43146.80

TABLE II
Timing

Transfer rule	Coefficients		
	a	b	c
GLS	−8.44733	−0.19019	0.10286
LLS	−14.06760	1.90661	0
VLS	−10.72000	1.47534	0.03218

In Fig. 3 we have plotted the mean values of MaxMemoryUsed $\overline{m}$ as a function of the system size N. It is seen that the GLS rule is the least memory consuming one, and that memory requirement for the VLS rule is a dozen percent higher than for the LLS rule. The memory usage in bytes can be approximated by the linear function

$$\overline{m}(N) = aN + b \tag{2}$$

with coefficients a and b shown in Table I.

Linear dependence observed for the memory usage is not observed for the CPU time consumption. Recorded values of both, the Timing and the AbsoluteTiming, functions have almost equal mean values and the mean CPU time consumption can be approximated by the function:

$$\overline{t}(N) = \exp(a + b \ln N + c \ln^2 N) \tag{3}$$

which reduces to

$$\overline{t}(N) = e^a N^{b + c \ln N} \tag{4}$$

Fitted values of coefficients a, b and c are presented in Table II. From this Table we see that the LLS algorithm is characterised by polynomial time complexity ($c = 0$) whereas the GLS and VLS algorithms have quasi-polynomial time complexity ($c \neq 0$).

Based on large data sets collected during simulations we have obtained and analysed empirical distribution of memory usage as well as a distribution of CPU time consumption. As a result we have obtained a very low dispersion of the memory distribution. For the CPU time consumption only the LLS

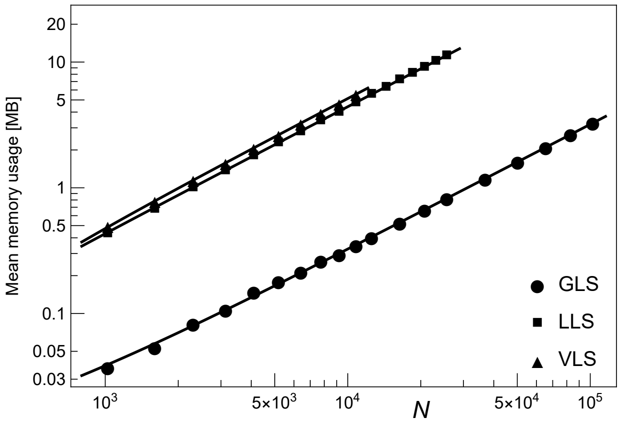


Fig. 3. Log-log plot of memory usage for GLS, LLS and VLS.

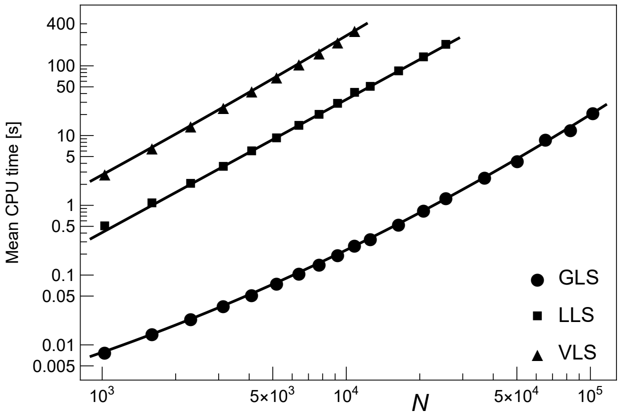


Fig. 4. Log-log plot of CPU time consumption – GLS, LLS and VLS.

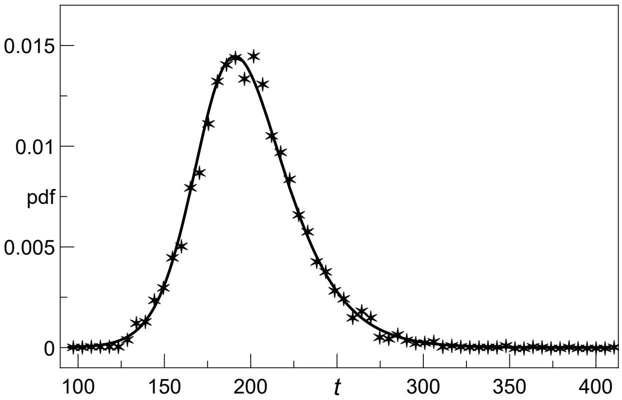


Fig. 5. Empirical probability density function (pdf) of AbsoluteTiming for arrays with $N = 160 \times 160$ and LLS. The solid lines represent Meixner distributed t with the parameters computed from the simulations.

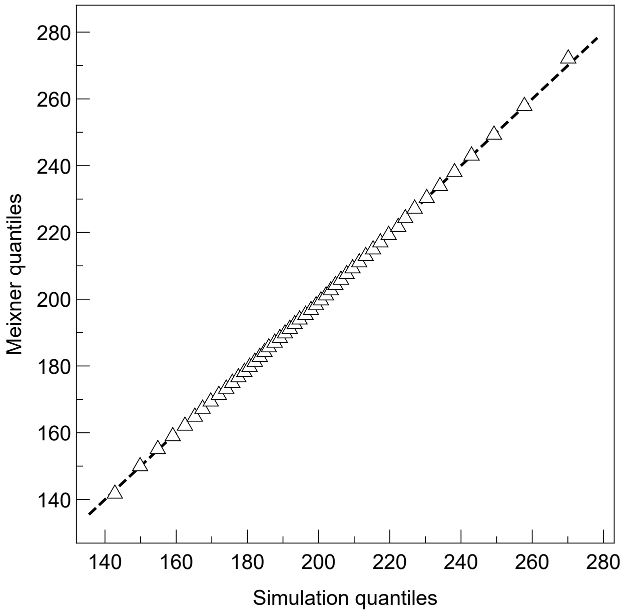


Fig. 6. The Q-Q plot of the quantiles of the set of the LLS AbsoluteTiming vs the quantiles of the Meixner distribution. System size $N = 160 \times 160$.

AbsoluteTiming has the distribution with regular shape. In this particular case the resulting distribution follows Meixner distribution well known as a model of financial data [12]). The probability density of this function is given by

$$p_N(t) = \frac{(2\cos(s/2))^{2\lambda}}{2k\pi\Gamma(2\lambda)} \exp\left(\frac{s(t-t_0)}{k}\right) \left|\Gamma\left(\lambda + \frac{i(t-t_0)}{k}\right)\right|^2 \qquad (5)$$

where: t_0 – location parameter, $k > 0$ – scale parameter, $-\pi < s < \pi$ – skew parameter and $\lambda > 0$ – shape parameter. Fig. 5 shows empirical probability density function of AbsoluteTiming for the LLS system of 160×160 pillars. In this plot we have also added fitting lines of Meixner probability density function with parameters computed from the samples. We also present a quantile-quantile plot (see Fig. 6) of the quantiles of the collected data set against the corresponding quantiles given by the Meixner probability distribution.

TABLE III
COEFFICIENTS OF (6) FOR SCALED LLS COUNTERS

Counter	**Coefficients**			
	x	μ	y	γ
LT/N	2.93671	-0.5	0.17202	-0.05259
LI/N	0.50546	-0.5	0.16263	-0.05671
OLT/N	0.37869	-0.5	2.01440	-0.00494
NNS/N	-4.16434	-0.5	4.00116	0
FNS/N	6.33348	0	0.00544	0.56637
OLI/N	0	0	0.18801	0.92690

A closer look at complexity of the algorithm can be gain introducing the following counters within the code:

- LT – the number of sequences of load transfers.

- LI – the number of steps of external load increases.
- OLT – the overall number of load transfers. OLT counts all load transfers to individual pillars.
- NNS – the number of nearest neighbour's searches.
- FNS – the number of further neighbour's searches.
- OLI – the overall number of external load increases on individual pillars.

In order to get intensive-like estimates of these counters we have scaled their mean values by the number of pillars N. It turns out that after the scaling all the counters are nicely approximated by the following function:

$$CV(N) = xN^{\mu} + yN^{\gamma} \tag{6}$$

Fitted values of x, y, μ, γ are reported in Table III. Exemplary counters with adequate fittings are shown in Figs. 7-9.

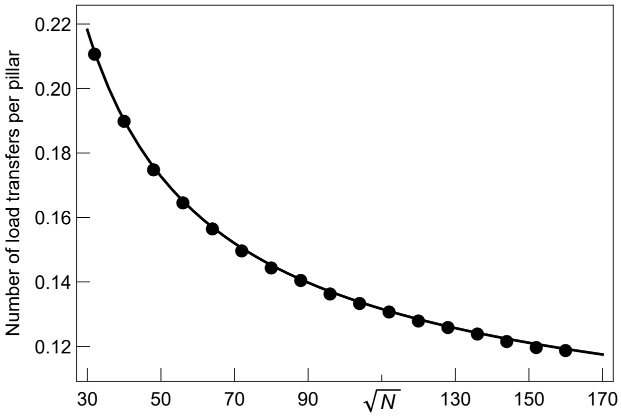


Fig. 7. Scaled number of load transfers LT/N as a function of $\sqrt{N}$.

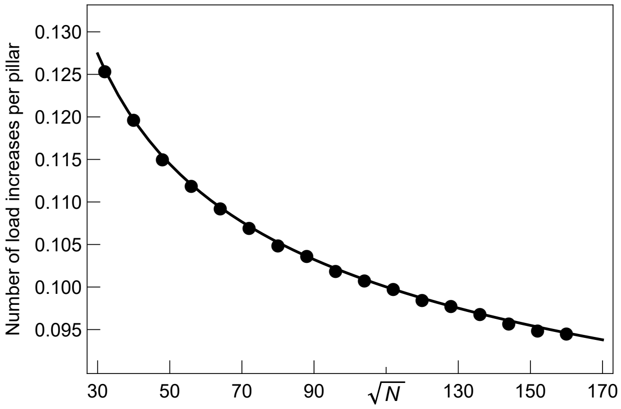


Fig. 8. Scaled number of load increases LI/N as a function of $\sqrt{N}$.

The counters LT, LI and OLT decrease with N, which means that their influence on computation time become marginal. Nevertheless, LT and LI represent two main sections of the code. Depending on the system size N, different numbers of operations are processed within sections of our code. These numbers of operations, represented by such counters as OLT, NNS, FNS and OLI, do not scale uniformly with N. When analysing scaled values of these counters, it turns out that three of them: NNS, FNS and OLI increase with growing N. This indicates that internal operations represented by these counters are more significant than the operations represented by LT and LI. As a consequence the computation time increases nonlinearly with N. Especially, the counter OLI can serve as a dominant variable in the regression model of the scaled mean computation time (in seconds):

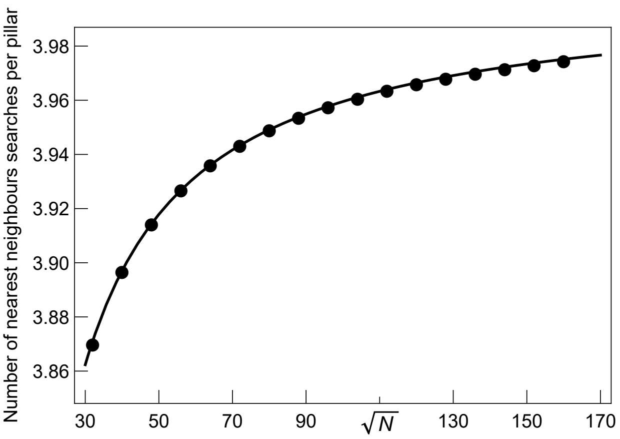


Fig. 9. Scaled number of nearest neighbours searches NNS/N as a function of $\sqrt{N}$.

$$\bar{t}(N)/N \times 10^3 = \delta + \kappa \times OLI + \theta \times LT \tag{7}$$

with $\delta = -0.42129, \kappa = 0.00374, \theta = 2.25441$, as is shown in corresponding Fig. 10.

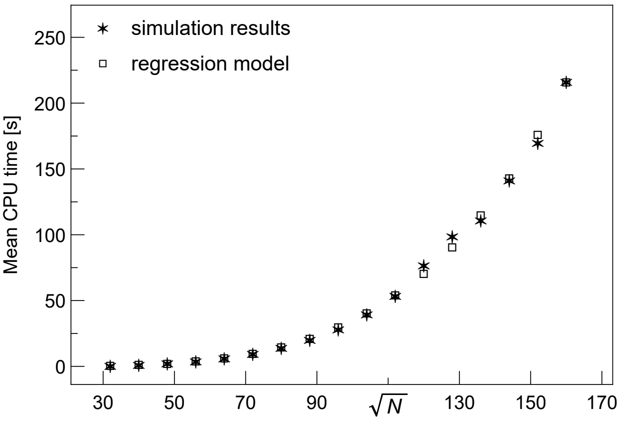


Fig. 10. Mean CPU time: simulation results (stars) and regression model (squares) vs. $\sqrt{N}$.

To summarise, we carried out simulations of damage evolution in progressively loaded arrays of pillars being a two-dimensional systems that simultaneously undergo series of component's failures and consecutive load redistributions. We considered components placed in nodes of a square lattice and characterized by quenched random components-load-thresholds drawn from the Weibull probability distribution. We computed and analysed amounts of memory m and CPU time consumption t during numerical experiments conducted within the Mathematica v.11 environment. Based on the presented results of simulation study we conclude that the experimental distributions of the allocated memory m as well as the CPU time consumption t can be effectively estimated. By fitting discrete distributions, we have found that: (i) $\overline{m}$ scales linearly with the number of system components N, whereas (ii) $\bar{t}$ exhibits a quasi-polynomial behaviour.

References

[1] A. Hansen, P.C. Hemmer, and S. Pradhan, "The Fiber Bundle Model: Modeling Failure in Materials," Wiley, 2015.

[2] S. Pradhan, A. Hansen A., and B.K. Chakrabarti, "Failure processes in elastic fiber bundles," *Rev. Mod. Phys.*, vol. 82, pp. 499-555, 2010.

[3] M.J. Alava, P.K.V.V. Nukala, and S. Zapperi, "Statistical models of fracture," *Adv. In Physics*, vol. 55, pp. 349-476, 2006.

[4] F. Kun, F. Raischel, R.C. Hidalgo, and H.J. Herrmann, "Extensions of fibre bundle models," *Modelling Critical and Catastrophic Phenomena in Geoscience, Lecture Notes in Physics*, vol. 705, pp. 57-92, 2006.

[5] Z. Domański, T. Derda, and N. Sczygiol, "Critical avalanches in fiber bundle models of arrays of nanopillars," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2013, IMECS 2013*, 13-15 March, 2013, Hong Kong, pp. 765-768.

[6] Z. Domański and T. Derda, "Distribution of critical load in arrays of nanopillars," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2017, WCE 2017*, 5-7 July, 2017, London, pp. 797-801.

[7] Z. Domański, T. Derda, and N. Sczygiol, "Statistics of critical avalanches in vertical nanopillar arrays," in: Yang GC., Ao SI., Huang X., Castillo O. (eds) *Transactions on Engineering Technologies. Lecture Notes in Electrical Engineering*, vol 275, pp. 1-11, Springer, Dordrecht, Dec. 2013.

[8] W.I. Newman and S.L. Phoenix, "Time-dependent fiber bundles with local load sharing," *Phys. Rev. E* 63, 021507, January 2001.

[9] S.L. Phoenix and W.I. Newman, "Time-dependent fiber bundles with local load sharing. II. General Weibull fibers," *Phys. Rev. E* 80, 066115, December 2009.

[10] N.M. Pugno and R.S. Ruoff, "Nanoscale Weibull statistics," *J. Appl. Phys.*, vol. 99, id. 024301, 2006.

[11] W. Weibull, "A statistical distribution function of wide applicability," *J. Appl. Mech.*, vol. 18, pp. 293-297, 1951.

[12] W. Schoutens,"The Meixner Process: Theory and Applications in Finance," Eindhoven : Technische Universiteit Eindhoven, February 2002.

Estimating the time complexity of the algorithms by counting the Java bytecode instructions

Tomaž Dobravec
Faculty of Computer and Information Science
University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenija
Email: tomaz.dobravec@fri.uni-lj.si

***Abstract*—Ranging the selected algorithms with the same theoretical time complexity boundaries can only be done by comparing the behavior of their implementations in the real environment. Timing the algorithms in practice is very difficult since it is hard to ensure a fair and reproducible environment in which implementations can be compared. In this paper we present a system called ALGATOR that was developed to facilitate the process of testing, comparing and evaluating the algorithms. Besides the time complexity indicators, ALGATOR also measures the usages of the Java bytecode instructions. We present the usage of the ALGATOR's JVM indicators for the estimation of the time complexity of selected bytecode instructions. We also present a method to predict the behavior of the simple algorithm for matrix multiplication using the results of the JVM indicators analysis.**

I. Introduction

The results obtained by the last two phases of the algorithm design process (i.e. the proof of the correctness and the estimation of the complexity) are based on the assumptions of the selected computational model, which (more or less successfully) imitates the real environment in which the algorithm will be implemented. From a theoretical point of view, these results are interesting and completely legitimate, since they allow (theoretically) to compare and classify algorithms. When applying these results in practice, however, problems may arise, as it often shows that the real environment differs from the theoretical assumptions. Thus, for example, theoretical models usually do not include assumptions about the concrete implementation of memory management (and more specifically, the influence of the cache), which in practice greatly affect the speed of implementation.

To choose the best algorithm for solving a given problem, theoretical results may help in the first round of selection, where the algorithms with the best asymptotical boundaries are selected. A real distinction between the selected algorithms with the same theoretical boundaries can only be made by comparing their behavior in the real environment. Timing the algorithms in practice is very difficult since it is hard to ensure a fair and reproducible environment in which algorithms can be compared. The results of the measurements are influenced by various factors, some of them are more or less random. In order to better assess the practical time complexity, we, therefore, need additional tools to measure independent indicators of the implementation of algorithms.

In this paper we present a novel approach for measuring and predicting the complexity of the algorithms' implementations by counting the Java bytecode instructions. In Section II. we present a tool called ALGATOR [1], which was designed to facilitate the algorithm comparison process by measuring different indicators. We also present the three types of measurements which are supported by the ALGATOR. In Section III. we focus on a simple problem of matrix multiplication and present different approaches to produce useful performance predictions based on the java bytecode instructions usages. We conclude this paper with the final remarks in Section IV.

II. The ALGATOR

The ALGATOR is a computer application that was developed to support and to facilitate the algorithm design and evaluation process. The main entity within the system is the so called 'project' which it defined by a set of definitions for the problem, the algorithms and the test sets. The system was designed to be as general as possible and therefore applicable in a wide range of problem domains. All the entities (projects, algorithms, test cases) are primarily defined on an abstract level and the system is trained to execute abstract algorithms on abstract test cases. After selecting a real problem, user concretizes the abstract parts of the project and makes the project 'alive' and prepared to be used to execute real algorithms on real data.

The abstraction of the project is integrated in several parts of the ALGATOR system. The algorithm is defined as a block of code (e.g. a class in Java) with predefined hook used to pass the parameters and start the execution (e.g. a method signature). The test case and the results of the execution are defined as an arbitrary data structures implemented in a selected programming language. A test set, which represents a minimal execution unit, is composed of several test cases and it is iterated through during the algorithm execution process. To collect and present the results of the algorithm execution, the ALGATOR uses the so called result sets (i.e. the sets of parameters and indications of the execution) and the presenters (the definition files in which the type and the range of the presentation is provided). All these abstractions make the system flexible and usable in many fields of computer science.

A. The ALGATOR *project*

To define an ALGATOR project, user must provide both, the configuration files and the source code in a selected

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

programming language (Java, C or C++). The configuration files define administrative data (the name and the author of the project, the number of supported algorithms, the time limit for algorithm execution, ...) while the source code provide the logic for executing the algorithms and for evaluating the results of the execution. The configuration files use the `json` format and have predefined names and positions in the folder hierarchy of the project. For example, the configuration file for the project `P` is named `P.atp` and it is placed in the subfolder `proj` of the project folder `PROJ-P`. Besides the basic information about the project the configuration of the project also include the information about algorithms (the name and the author of each algorithm, programming language of the algorithm, ...), about test sets (the number of test cases of each test set, sizes of the test cases, time limits for execution of a test case, ...) and about the results (the number and the type of the indicators of execution).

The source code of the project is provided in the following classes (in the case of Java programming language; for C/C++ the logic is similar): `TestCase` (a class with predefined data structures needed to present the input and the output of the algorithms), `AbsAlgorithm` (a class that defines an abstract method that it will be used (when implemented) as the heart of the algorithm) and `TestSetIterator` (a bridge between the definition files and Java data structures; in this class test set configuration file is read and Java test case is generated).

B. Types of the ALGATOR *engines and users*

The ALGATOR was developed to be used as a standalone and/or as a server application. A standalone application is used to develop, test and evaluate algorithms in a separated domain where the results are used only by a limited and typically small group of people. This application can be installed and used on every personal computer. The main drawback of using the standalone version is that the results of the execution can not be fairly compared with the results of other groups of the researchers. On the other hand, the server application offers the possibility to run algorithms provided by different researchers of different groups on a single computer. The results obtained are accurate and comparable. The server version is usually installed on an internet server computer and accessed through the web interfaces.

ALGATOR supports four different user roles: the system administrator (installs and manages the whole system and has the access to all the resources of the system), the project administrator (defines the project and has an access to all the project resources), researcher (defines an algorithm for the selected project, runs predefined tests and compares the results with the results of other algorithms) and the guest (observes all public projects, algorithms, and test results). The logic of the user rights and roles is equally supported in both versions of the ALGATOR, but it is a bit relaxed in the standalone version since all the roles are usually played by a single user.

C. The measurements

For each problem there are several different measurements that ensure the correctness and by which one can assess the efficiency of the algorithms. These measurements include the indicators of time consumption and of the quality of the result, counters for the usage frequency of the parts of the program code, and the counters of the usage of the basic execution operations (i.e. the machine instructions). In the ALGATOR system all kind of the measurements are supported and are grouped into three categories: the EM, CNT and JVM indicators.

The EM indicators. The EM indicators are used to measure the time and other project-specific metrics . All measurements of the time indicators are performed automatically. To provide as accurate time indicators as possible the ALGATOR tries to reduce the influence of the uncontrolled computer activities (e.g. sudden increase of a system resource usage) by running each algorithm several times. The system measures the first, the best, the worst and the average time of the execution. The project administrator only needs to specify the phases of algorithm execution (e.g. the pre-processing phase, the main phase, the post-processing phase, ...) and to select which of the time indicators are to be presented as the result of execution. The project-specific indicators are defined by the project administrator. They can be presented as a string or as a number. For example, for exact algorithms, the value of an indicator could be ”OK” (if the algorithm produced the correct result) or ”NOK” (if the result of the algorithm is not correct). For approximation algorithms the value of an indicator could be the quality of the result (i.e. the quotient of the correct result and the result of the algorithm).

The values of the EM indicators are generated by ALGATOR performing the following steps:

1) load the test case and create its project-specific representation,
2) load the algorithm (if the algorithm is implemented in Java, for example, the ALGATOR uses the Java reflection capabilities to create the algorithm instance),
3) read the values of the test case specific parameters,
4) run the algorithm and measure the time consumption,
5) read and store the values of the time indicators,
6) determine and store the values of the project-specific indicators,
7) write stored indicators into the output as prescribed in the result description configuration files.

The CNT indicators. The CNT indicators (the so called *counters*) are used to count the usage of the parts of the program code. This option is used to analyze the usage of a certain system resource or to count the usage of the selected type of commands on the programming language level. Using this, one can, for example, measure how many times the memory allocation functions were executed during the algorithm execution and the amount of the memory allocated by these calls. One can also use CNT indicators to detect which part of the algorithm is most frequently used. For example, if the problem in concern would be the data-sorting, using the CNT indicators one could count the number of comparisons, the number of swaps of elements and the number of recursive function calls (which are the measures that can predict the algorithm execution behavior [2]). To facilitate the CNT indicators in the project, the project administrator has to define the names and the meaning of the counters and the researchers have to tag the appropriate places in their code. Everything else is done automatically by the ALGator.

The JVM indicators. Before the execution of the algorithm, the algorithm code has to be compiled to a code on a lower level. For the C and C++ projects this means that the algorithms are compiled to the machine code of the architecture used by the system, while for the Java projects this means that the algorithms are compiled to the Java bytecode. The performance of the algorithm depends on the number and the type of the low-level instructions used during the execution [3]. The ALGator enables the analysis of the low-level instruction usages for the algorithms written in the Java programming language. During the execution of the algorithm ALGator counts the bytecode instructions that were used and at the end it prints out the statistics for each instruction (the so called JVM indicators). To enable this facility, a special JVM (called VMep [4]) was developed based on the open-source Java Virtual Machine JamVM [5]. This facility makes ALGator very powerful tool for deep analysis of the algorithms' behavior. In the rest of this paper we will describe a basic example of how the JVM indicators can be used in practice. We will show an efficient method for predicting the time consumption based on the analysis of the JVM indicators.

III. Using the JVM indicators in practice

To explore the measuring capabilities of the ALGator we chose a simple matrix multiplication problem: given two square matrices A and B each containing n^2 elements (a_{ij} and b_{ij} for $i, j = 0, \ldots, n-1$) calculate the elements of a square matrix C by

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} b_{kj}.$$

Since the number of operations in this formula is cubic to n, we can reasonably expect that the time complexity of any algorithm implementing this formula would have the time complexity $\Theta(n^3)$. The simple implementation of this formula is presented in listings in Figure 1.

```
void MUL(int[][] A, int[][] B, int[][] C) {
  for (int i = 0; i < A.length; i++) {
    for (int j = 0; j < A.length; j++) {
      for (int k = 0; k < A.length; k++) {
        C[i][j] += A[i][k] * B[k][j];
      }
    }
  }
}
```

Fig. 1. The Java code for the MUL algorithm

```
void MUL(int[][], int[][], int[][]);
   0: iconst_0       |    36: dup2
   1: istore 4       |    37: iaload
   3: iload 4        |    38: aload_1
   5: aload_1        |    39: iload 4
   6: arraylength    |    41: aaload
   7: if_icmpge 73   |    42: iload 6
  10: iconst_0       |    44: iaload
  11: istore 5       |    45: aload_2
  13: iload 5        |    46: iload 6
  15: aload_1        |    48: aaload
  16: arraylength    |    49: iload 5
  17: if_icmpge 67   |    51: iaload
  20: iconst_0       |    52: imul
  21: istore 6       |    53: iadd
  23: iload 6        |    54: iastore
  25: aload_1        |    55: iinc 6, 1
  26: arraylength    |    58: goto 23
  27: if_icmpge 61   |    61: iinc 5, 1
  30: aload_3        |    64: goto 13
  31: iload 4        |    67: iinc 4, 1
  33: aaload         |    70: goto 3
  34: iload 5        |    73: return
```

Fig. 2. The Java bytecode for the MUL algorithm

We named this implementation the MUL algorithm since its main (and the most consumptive) operation is the multiplication. Using the ALGator's time complexity indicators we measured the time needed to execute this algorithm on a set of test cases with dimensions n ranging from 200 to 500. We run all the tests described in this paper on a personal computer with Intel(R) Core(TM) i7-6700 CPU running at 3.40GHz with 32Gb of memory. The execution of the matrix multiplication for smaller inputs (n=200) was done in 6000 microseconds and for larger matrices (n=500) in 140.000 microseconds. To eliminate the impact of the real environment we executed all the tests (i.e. we calculated each product) for 500 times and we took the minimal time of all executions (obviously, this is the time in which the execution can be performed if the environmental influences are as small as possible). The time of the execution for the matrix multiplication problem is depicted in Figure 3 with blue line. In the same graph a simple prediction for the execution time is also depicted. It was calculated by a simple method called Calc1. In this method we calculated a multiplication factor $c = avg(time_i/n^3)$. The red dots in Figure 3 represents a graph of a function cn^3. Obviously the red dots are of the same shape as the blue line (which is due to the fact that our altorithm has $\Theta(n^3)$ time complexity) but it is not very accurate. An average error (i.e. the difference between measured (blue) and calculated (red) time divided by measured time) is 11,25%. This error is a bit smaller (i.e 7,1%) if we take only bigger dimensions of input matrices (from 300 to 500), but it is still relatively big. Therefore the method Calc1 can not be considered a very

n	ICONST_0	ILOAD	ALOAD_1	ALOAD_2	ALOAD_3	IALOAD	AALOAD	ISTORE	IASTORE	DUP2	IADD	IMUL	IINC	IF_ICMPGE	GOTO	ARRAYLENGTH
10	111	7221	2221	1000	1000	3000	3000	111	1000	1000	1000	1000	1110	1221	1110	1221
20	421	56841	16841	8000	8000	24000	24000	421	8000	8000	8000	8000	8420	8841	8420	8841
30	931	190861	55861	27000	27000	81000	81000	931	27000	27000	27000	27000	27930	28861	27930	28861
40	1641	451281	131281	64000	64000	192000	192000	1641	64000	64000	64000	64000	65640	67281	65640	67281
50	2551	880101	255101	125000	125000	375000	375000	2551	125000	125000	125000	125000	127550	130101	127550	130101

TABLE I
THE JAVA BYTECODE INSTRUCTIONS USAGES IN THE MATRIX MULTIPLICATION ALGORITHM.

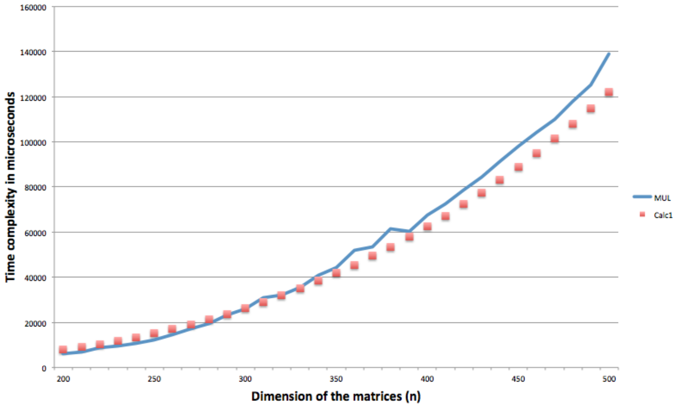


Fig. 3. The time complexity of the MUL algorithm (blue line) and the performance prediction calculated by a simple Calc1 method (red dots).

successful method.

In order to find better performance prediction for this algorithm we used ALGATOR's capability for measuring the usages of the Java bytecode instructions (the bytecode for the algorithm MUL is listed in Figure 2). The results show that only 16 (out of 202) Java bytecode instructions are used during the execution of this algorithm: 10 instructions for the stack manipulation (ICONST_0, ILOAD, ALOAD_1, ALOAD_2, ALOAD_3, IALOAD, AALOAD, ISTORE, IASTORE, DUP2), two instructions to control the flow of the program (IF_ICMPGE, GOTO), the ARRAYLENGTH instruction used to determine the size of an array, and three arithmetic instructions (IADD, IMUL, IINC). The frequencies of the usages of these instructions for the matrices of sizes from 10 to 50 are presented in Table I. As it is clearly seen from the data in the table, for most of the instructions their usages in the matrix multiplication algorithm is of the order $\Theta(n^3)$. The only exceptions are the instructions ICONST_0 and ISTORE with the order $\Theta(n^2)$. From the data presented in Table I we calculated the overall number of the instructions $\mathtt{INST}(n)$ used in the MUL algorithm:

$$\mathtt{INST}(n) = 25n^3 + 12n^2 + 12n + 6.$$

This means that in the case of $n = 500$, for example, the JVM performs $25 \times 500^3 + 12 \times 500^2 + 12 \times 500 + 6 = 3.128.006.006$ bytecode instructions to execute the MUL algorithm. Since this execution requires approximately 140.000 microseconds, an average time to execute one Java byte code instructions is 0,044ns.

Analyzing the results presented in Table I (extended with measurements for n=60, ..., 500) a natural question arises: can we calculate an average time (over all the measurements) used to execute one bytecode instruction and use this average to predict the behavior (i.e. time consumption) of the MUL algorithm for a given n. To find an answer to this question, we propose the following method Calc2: calculate the average time I_n used for one bytecode instructions while performing MUL on the matrix of size n (e.g. $I_{500} = 0,044$) and calculate I as an average of I_n. Then use I to estimate the execution time of MUL by $T(n) = I * INST(n)$. Using this method we calculated $I = 0,039ns$ (note that we used only measurements for $n = 300, ..., 500$ since we assume that the measured times are much more accurate for bigger inputs). Surprisingly, the Calc2 method gives very similar results as the method Calc1: an average difference between those two methods is 0,03% for $n = 300, \ldots, 500$. In other words, calculating the uniform average time per bytecode instruction yields another useless method for estimation of time consumption.

The main reason for bad results is that some bytecode instructions are much more expensive than the others. For example, we can reasonably assume that the IMUL instructions takes much more time to execute than the ILOAD instruction (the first instruction multiplies two integers while the second one loads an integer onto a stack). The question is, how many different types of instructions (instructions of the same type take approximately the same time to execute) are included in the MUL algorithm. To answer this question we implemented two algorithms, both of then very similar to MUL. The first one, the ADD algorithm, is an exact copy of MUL with the only difference in the line 5 where we instead of multiplication use addition (`C[i][j] += A[i][k] + B[k][j];`). The execution of this algorithm results in the usages of exactly the

```
void SET(int [][] A, B, C) {
  int x=0,y;
  for (int i = 0; i < A.length; i++) {
    for (int j = 0; j < A.length; j++) {
      for (int k = 0; k < A.length; k++) {
        y = A[i][j];
        B[i][j] = x;
        C[i][j] = y;
        x++;
      }
    }
  }
}
```

Fig. 4. The Java code for the `SET` algorithm

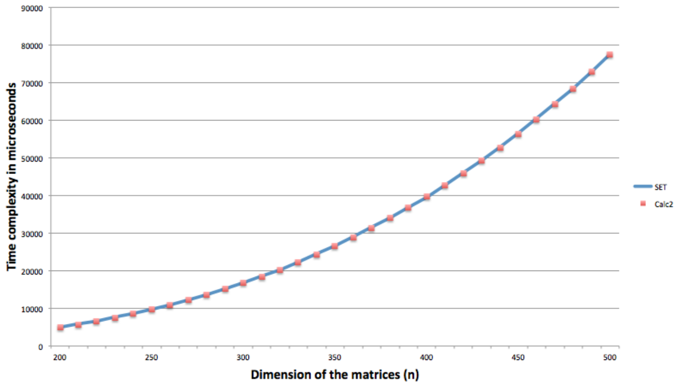


Fig. 5. The time complexity of the `SET` algorithm (blue line) and performance prediction calculated by a `Calc2` method (red dots).

same Java bytecode instructions, the only difference is that instead of IMUL in `ADD` algorithm only IADD instruction is used (which is logical, but we also proved this by scanning the ALGATOR's jvm indicators). As a consequence, in the `MUL` we have n^3 IADDs and n^3 IMULs while in `ADD` we have only $2 \times n^3$ IADDs. The number of all the other instructions is equal in both algorithms. In the second algorithm, `SET`, we deleted line 5 of `MUL` and replaced it with 4 lines as showed in the listings in Figure 4.

The resulting `SET` algorithm compiles into a Java bytecode program with exactly the same number of instructions as the `MUL`, which means that for executing `SET` on matrices of size n, JVM also performs $INST(n)$ bytecode instructions. The only difference is that the `SET` does not use the IADD and IMUL instructions.

The algorithm `SET` uses only the following twelve instructions: ICONST_0, ILOAD, ALOAD_1, ALOAD_2, ALOAD_3, IALOAD, AALOAD, ISTORE, IINC, IF_ICMPGE, GOTO, ARRAYLENGTH. We made an assumption that these instructions are all equally consumptive, we named them as "simple instructions", and we used the method `Calc2` to calculate their average execution time I. Using the resulting $I = 0,0248ns$ (for the further reference we will denote it with I_s) and formula $T_{SET}(n) = I_S * INST(n)$ we found out that the `Calc2` methods in this case yields almost a perfect estimation. Figure 5 shows the measured time of the `SET` method (blue line) and its estimation provided by `Calc2` method. An average error ($n = 200, \ldots \ldots, 500$) of this method is 0,4%. This means that the calculated $I_s = 0,0248$ nanoseconds is a reasonably good estimation for the execution time of every simple Java bytecode instruction on this computer.

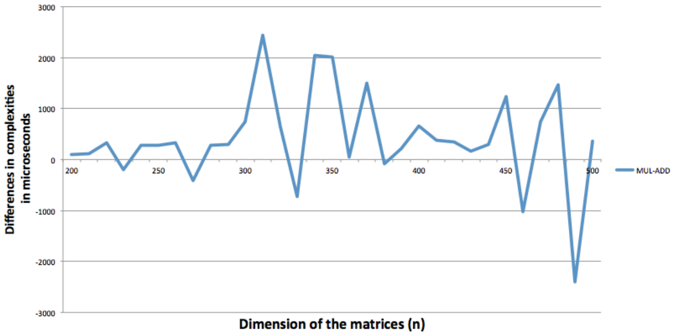


Fig. 6. The differences in the time complexities of the `MUL` and `ADD` algorithms.

To make a good estimation for the `MUL` algorithm we now only have to determine the estimation for the time complexities of the IMUL and IADD instructions. First we compare the execution time of the algorithms `MUL` and `ADD` to find out that these two algorithms are comparable in the sense of time consumption. Graph 6 shows the differences in the time complexities of the `MUL` and `ADD` algorithms. According to the oscillation of the graph we can conclude that both algorithms are equally consumptive, the repetitive exchange of the leadership (positive and negative values on the graph) indicates that the times of execution were measured with a certain relatively small (on average less than 1%) error. Since the only difference between `MUL` and `ADD` is in the number of IMUL and IADD instructions used (the second one uses only IADDs while the first one uses both) and since we proved that there is no real difference in time consumption, we can conclude that the time consumptions of the IADD and IMUL instructions are the same. This is not just an interesting result but it also gives us an opportunity to estimate the real time consumption of both arithmetic instructions. In the `MUL` we have $INST(n)$ instructions among which there are $2 \times n^3$ arithmetic instructions. Assuming that the time complexity of an arithmetic instruction (I_A) equals $I_A = I_S + \lambda$ we get

$$\lambda = AVG_n \left(\frac{T_{MUL}(n) - T_{SET}(n)}{2 * n^3} \right).$$

Using the measured times of `MUL` and `SET` and averaging for $n = 300, \ldots, 500$ we obtain $\lambda = 0,22$ and $I_A = 0,24$ nanoseconds. This means that an average cost of an arithmetic operations IADD and IMUL is 9,7-times bigger than an average cost of a simple instruction.

To estimate the execution time of the `MUL` algorithm we use the following `Calc3` method: given the factors I_S and I_A,

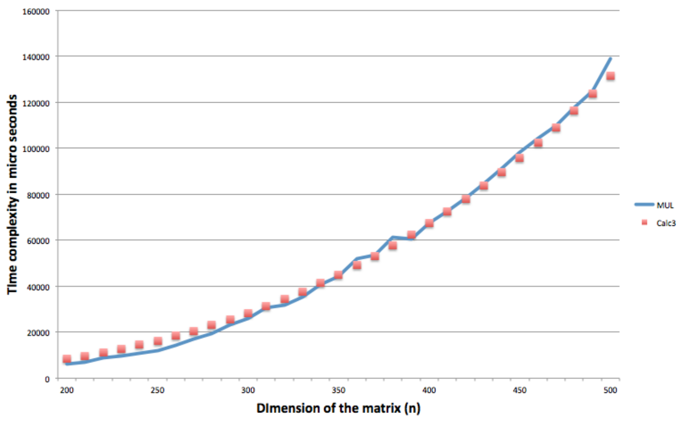


Fig. 7. The time complexity of the `MUL` algorithm (blue line) and performance prediction calculated by a `Calc3` method (red dots).

calculate the estimation of the time complexity of the `MUL` algorithm by

$$T_{MUL}(n) = (INST(n) - 2n^3) \times I_S + 2n^3 \times I_A.$$

Using this estimation we find out that it much better fits the `MUL` algorithm then the previous ones. Graph in Figure 7 shows the time complexity of `MUL` with blue line and the `Calc3` estimation with red dots. An average error of this estimation (for $n = 300, \ldots, 500$) is 2.3%.

IV. Conclusions

In this paper we described the ALGator – a system for testing and analyzing the algorithms. We showed how to use the ALGator's ability to count the usages of the Java bytecode instructions. Using three algorithms (`MUL`, `ADD` and `SET`) we presented different (more or less efficient) methods to produce the performance prediction of the algorithms based on the number of Java bytecode instruction used.

We showed that the "simple" instructions (e.g. ILOAD_0, IALOAD, ISTORE, ...) are equally time consumptive and that they on average take 0,0248 nano seconds to execute (on our computer). We also showed that arithmetic instructions (IADD and IMUL) are much more time consumptive - on our computer these instructions take 0,24 nano seconds (which is almost 10 times slower than the simple instructions). Using these information about bytecode instructions and the formula for total bytecode instruction usages (which was also derived from the results of ALGator's execution) we presented a method for the execution time prediction of the selected algorithm for matrix multiplication. The results of this method were much better then the results of a basic (naive) method which estimates the time complexity with a simple cubic function.

The ALGator's capability to count the Java bytecode usages helps us to better understand the behavior of the algorithms. The test case presented in this paper is very educative, but it is not general because of the nature of the selected algorithm (the behavior of the algorithm is totally deterministic and does not dependent on the input data; the algorithm always uses the same instructions regardless the content of the input matrices). To prove a general usability of the JVM indicators other problems and algorithms should be concerned.

References

[1] T. Dobravec, "The algator system." github.com/ALGatorDevel/Algator, 2012-2017.

[2] R. Segedwick, "The analysis of quicksort programs," *Acta Informatica*, vol. 7, pp. 327–355, 1977.

[3] J. M. Lambert and J. F. Power, "Platform independent timing of java virtual machine bytecode instructions," *Electronic Notes in Theoretical Computer Science*, vol. 220, pp. 79–113, 2008.

[4] J. Nikolaj, "Predelava javanskega navideznega stroja za štetje ukazov zložne kode," *Univerza v ljubljani, Fakulteta za računalništvo in informatiko, diplomsko delo; github.com/nikolai5slo/jamvm*, 2014.

[5] R. Lougher, "Jamvm," jamvm.sourceforge.nett, 2014.

The variability constraints in simulation of index-2 differential-algebraic processes

Paweł Drąg
Department of Control Systems and Mechatronics
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Email: pawel.drag@pwr.edu.pl

Krystyn Styczeń
Department of Control Systems and Mechatronics,
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Email: krystyn.styczen@pwr.edu.pl

***Abstract*—In the article a general simulation task of index-2 differential-algebraic equations (DAEs) is considered. To overcome difficulties connected with a nonlinear process dynamics, a new constraints type - the variability constraints - was introduced. The designed solution procedure is based on a multiple shooting method - it was assumed, that process dynamics in each shooting interval is constant. Therefore, the presented methodology can be treated as an extended direct shooting approach for simulation of the index-2 DAE equations with the variability constraints, which have an important practical application in various areas of applied informatics, e.g. manufacturing process modeling, simulation and optimization. Finally, the presented methodology was investigated on a two-phase reactor simulation task, which was described by the index-2 differential-algebraic equations.**

***Index Terms*—process simulation, differential-algebraic systems, variability constraints.**

I. Introduction

The time period of the past two decades can be characterized by a significant advance in optimization and control of systems modeled by pure dynamical equations (ordinary differential equations, ODEs), as well as processes described by differential-algebraic equations (DAEs). Especially, a progress in applied informatics on manufacturing processes modeling by differential-algebraic equations can be observed in new approaches in numerical analysis [7], [16], specialized computer modeling tools [14], as well as computational environments equipped with efficient numerical solvers, which enable effective model simulations [15].

In general, the differential-algebraic models in engineering arise from the equation based modeling of physical systems and can be found in such specified fields and applications, as chemical distillation processes [3], [13], electronic circuits [17], as well as mechanical systems design [18].

A characteristic feature of DAE equations, which is mainly used to determine their classification, is a differentiation index. The differentiation index is equal to the number of times all or part of the system has to be differentiated in order to solve the problem as ODEs. It is well known that solving a high index (larger than one) DAEs directly is numerically difficult. In the simplest form, the pure dynamical system is DAEs system with differential index equal to zero [4].

Therefore, using the idea of the differentiation index, two main DAE equations categories can be distinguished. There are the DAE systems with index equal to one, and DAEs with index greater than one, referred to as high-index systems. Especially, index-one DAE equations have been well understood and described and in a some sense share similar computational properties with ODEs. In contrast to index-one DAEs, higher index DAEs demonstrate significant differences compared to the pure dynamical equations. This difference can be reflected in numerical algorithms design, where it is wanted to avoid differentiating computed quantities. Recently, an issue of the DAEs index is under intensive research area of computational informatics. Ratnakar and Balakotaiah [21] presented a bifurcation analysis for index infinity DAE parabolic models. The proposed approach has been applied to solve steady-state models of chemical reactors and reacting flows. Then, a new procedure for solving higher index DAE optimal control problems has been designed by Campbell and Kunkel [5]. The presented approach has been investigated on a nonlinear index three control problem. Zhang and Zhang [28] investigated a two-loop procedure for solving index-3 DAE dynamic problems. Finally, Zeng et al. [27] implemented and tested the Hungarian method on the structure index reduction for DAE equations.

The main difference between differential-algebraic and pure dynamical equations is the presence of algebraic constraints, which results in difficulties in consistent initial conditions specification [9]. The physical phenomenon such as mass and energy conservation laws, thermodynamic equilibrium relations, Kirchhoff laws for electrical circuits, as well as pseudo-steady state processes can be described by the algebraic equations. Then, if the manufacturing processes are modeled by the implicit and singular algebraic equations, then the reformulation the DAEs model as an ODEs may be difficult or even impossible [2]. Therefore, both differentiations, as well as algebraic manipulations have been used to rewrite the higher-index DAEs to index-one DAEs or ODEs. Moreover, the index-reduction approach is one of the most often used approaches for high-index DAEs simulations [6], [20], [23].

One of the most important group of differential-algebraic equations simulation methods is to transform differential-algebraic constraints into a nonlinear optimization problem. This methodology has been successfully applied in many real-life manufacturing processes in chemical engineering, such as

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

- an optimization of a fed-batch bioreactor for 1,3-propanediol production [24],
- optimization of load change operation of high-temperature gas-cooled reactor [12],
- nitroxide-mediated radical polymerization of styrene[29],
- an industrial gas phase polymerization [10],
- parameter estimation in low-density polyethylene tubular reactors [25], as well as
- a polyurethane production[26].

The recently expectations form industrial processes require to develop efficient industrial informatics algorithms for nonlinear DAE models optimization. Moreover, among the well-known constraints types, like constraints resulting from sources or technology, in the presented work also variability constraints are considered. The variability constraints have not been discussed in this context.

The objective of this work is to design the new methodology to simulate the processes modeled by index-2 nonlinear differential-algebraic equations extended by the variability constraints. The application of the proposed methodology has been demonstrated on a two-phase reactor system, wich is modeled by an index two differential-algebraic equations.

The presented work is constructed in the following form. In the Section 2 the DAE equations were shortly characterized and the main features of these systems were given. Then, in Section 3 the variability constraints were introduced as an extension of processes modeled with differential-algebraic constraints. The new computational optimization procedure for the DAE systems with variability constraints was designed in Section 4. The presented algorithm was applied to optimization a two-phase reactor. The presented considerations, as well as results of numerical simulations, were presented and summarized in Section 5.

II. Differential-algebraic equations

In recent years, the dynamical features of the real-life processes are often modeled by systems of differential-algebraic equations. Moreover, both differential, as well as algebraic equations can be used to describe different components of the considered process [22]. Therefore, the differential-algebraic equations can be expressed in a general form as

$$F(\dot{z}, z, u, y, p, t) = 0 \tag{1}$$

extended by an output equation

$$G(z, u, y, p, t) = 0, \tag{2}$$

where $z \in \mathcal{R}^{n_z}$ denotes a vector of state variables and $\dot{z} = \frac{dz}{dt}$. The control function has been denoted as $u \in \mathcal{R}^{n_u}$, $y \in \mathcal{R}^{n_y}$ represents an output value. The independent variable, often time, was denoted by $t \in \mathcal{R}$, and vector $p \in \mathcal{R}^{n_p}$ describes design parameters constant in the time. Moreover, two vector-valued functions are considered

$$F : \mathcal{R}^{n_z} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_y} \times \mathcal{R}^{n_p} \times \mathcal{R} \rightarrow \mathcal{R}^{n_k} \tag{3}$$

$$G : \mathcal{R}^{n_z} \times \mathcal{R}^{n_u} \times \mathcal{R}^{n_y} \times \mathcal{R}^{n_p} \times \mathcal{R} \rightarrow \mathcal{R}^{n_l} \tag{4}$$

such that

$$rank \quad \frac{\partial F}{\partial \dot{z}} = n_{z_D} \tag{5}$$

$$rank \quad \frac{\partial G}{\partial z} = n_{z_A} \tag{6}$$

and

$$n_z = n_{z_D} + n_{z_A}. \tag{7}$$

Therefore, in the mathematical model both differential, as well as algebraic state variables can be distinguished

$$z = [z_D \quad z_A] \in \mathcal{R}^{n_z} = \mathcal{R}^{n_{z_D}} \times \mathcal{R}^{n_{z_A}}. \tag{8}$$

In this work, the DAE equations of the following special form are under considerations

$$\begin{aligned} \dot{z}_D &= f_1(z_D) + f_2(z_D) z_A + f_3(z_D) u \\ 0 &= g_1(z_D) + g_2(z_D) z_A \\ y_i &= h_i(z_D), \quad i = 1, \cdots, m \end{aligned} \tag{9}$$

where $f_1(z_D)$ is a vector-valued function on $\mathcal{R}^{n_{z_D}}$, $g_1(z_D)$ is a vector-valued function on $\mathcal{R}^{n_{z_A}}$, $f_2(z_D)$, $f_3(z_D)$ and $g_2(z_D)$ are matrices of dimensions $(n_{z_D} \times n_{z_A})$, $(n_{z_D} \times n_u)$ and $(n_{z_A} \times n_{z_A})$, respectively. Finally, $h_i(z_D), i = 1, \cdots, m$, are scalar functions. The differential-algebraic equations were presented in the semi explicit form (9), with the appearing linearly algebraic variables.

As it was discussed in the work [11], the consideration of the semi explicit form is motivated by industrial processes specification. It is constituted by modeling approach, where the differential equations are obtained explicitly according to mass and energy dynamic balances. Then, equilibrium relations, as well as empirical correlations, can be described by algebraic equations. Moreover, for some processes, e.g. as multiphase reaction and separation systems with equilibrium phase, the linear occurrence of the algebraic state variables is often observed.

One of the most important feature of differential-algebraic equations in the index ν. According to the definitions presented in [4], [19], the index ν denotes the minimum number of times the algebraic equations have to be differentiated to obtain a set of ordinary differential equations for the algebraic variables. Application of this procedure for the eq. (9) denotes, that

$$z_A = -g_2(z_D)^{-1} g_1(z_D). \tag{10}$$

The algebraic equation (10) can be solved directly for the algebraic state variable, if

$$\det g_2(z_D) \neq 0. \tag{11}$$

Then, after exactly one differentiation of eq. (10), the state variable z_A can be expressed by a system of differential equations.

Therefore, if the matrix $g_2(z_D)$ is nonsingular, then the DAE equations (9) have the index $\nu = 1$. For the index-one differential-algebraic equations the following ODE model can be obtained

$$\begin{aligned} \dot{z}_D &= f_1(z_D) - f_2(z_D)g_2(z_D)^{-1}g_1(z_D) + f_3(z_D)u \\ y_i &= h_i(z_D) \end{aligned} \tag{12}$$

for $i = 1, \cdots, m$.

Otherwise, if the matrix $g_2(z_D)$ is singular, then the algebraic variable z_A cannot be directly defined by the algebraic equations. In this case, more differentiations are necessary to obtain a model consisted on the pure differential equations. Such models are characterized by a higher index $\nu \geq 2$.

According to the presented reasons, a higher-index DAEs models, as well as their possible application in industry, are under considerations.

III. The variability constraints

The main difference between method presented in this work and classical approaches, is the introduction of the variability constraints. In general, the variability of the state variables is defined by the right-hand side of the dynamical equation

$$\dot{z}_D = f(\cdot), \tag{13}$$

where $(\cdot)$ denotes all arguments necessary to calculate the value of $f(\cdot)$.

The mathematical models of some real-life systems need additional constraints on the state trajectory variability rate. It means, that changes of the state trajectory need to be restricted in a reasonable way. This approach has an important practical application. In the literature the mathematical model of an aircraft landing is considered [1]. In this case, the changes of the flight parameters cannot be to sharp. Then, this assumption results in an additional model equation

$$\dot{z}_D \leq c, \tag{14}$$

where $c \in \mathcal{R}^{n_{z_D}}$ denotes a limit value of the permissible rate of changes. The question was, how to introduce the constraint (14) into a computational solving procedure. The classical solution procedure was based on the direct shooting method [8]. In this procedure the control function, as well as initial values of the DAE equations, are parametrized according to the chosen multiple shooting intervals.

Let us assume, that the process duration time

$$t \in [t_0 \quad t_F] \tag{15}$$

was divided into n subintervals. Then, the duration time of each subinterval can be denoted as

$$t^i \in [t_0^i \quad t_F^i] \tag{16}$$

with $i = 1, \cdots, n$.

The initial value of the differential state variables $z_D(t_0^i)$ are treated as additional decision variables. Moreover, the new optimization-based problem can be extended by pointwise constraints, which preserve the continuity of the considered trajectories

$$z_D(t_F^{i-1}) = z_D(t_0^i) \tag{17}$$

for $i = 2, \cdots, n$. The control function can be chosen and parametrized according to the assumed subintervals. One of the possible control function parametrization approach is, that the control function in each subinterval is constant

$$u(t^i) = u^i = const \tag{18}$$

and treated as a vector of decision parameters.

The presented direct shooting method enables us to apply sequential solving procedure, where the state trajectories are obtained by implemented appropriate ODE/DAE *solvers*. Finally, the system of dynamical equations needs to be solved

$$\begin{aligned} \dot{z}_D(t^i) &= F^i(z_D(t^i), z_A(t^i), u(t^i), p, t^i) \\ 0 &= G^i(z_D(t^i), z_A(t^i), u(t^i), p, t^i) \end{aligned} \tag{19}$$

for $t^i \in [t_0^i \quad t_F^i]$ and $i = 1, \cdots, n$. The trajectory $z_D(t)$ is a sum of the trajectories obtained in all subintervals. It means, that the direct shooting procedure in the presented form only slightly preserve the numerical difficulties, which can be observed, if the functions $F^i(\cdot)$ and $G^i(\cdot)$ are highly nonlinear. In this case, when the DAE solver would fail, then the solving procedure results with no solution. Therefore, the modified direct shooting procedure with the variability constraints was introduced.

IV. The solution procedure

The simulation of the DAEs model is connected with three main difficulties

a) consistent initial conditions are unknown,
b) process dynamics can be highly nonlinear,
c) the index of DAEs system can be higher than 1.

To overcome the mentioned difficulties connected with the process simulation, the extended direct shooting approach was designed. The designed procedure was consisted on the following steps.

- **Step 1.**
  Choose a number of subintervals n and divide the simulation time in the following way
  $$t_0 = t_0^1 < t_F^1 = t_0^2 < \cdots = t_0^N < t_F^N = t_F, \tag{20}$$
  that
  $$t^i \in [t_0^i \quad t_F^i] \tag{21}$$
  for $i = 1, \cdots, n$.

- **Step 2.**
  Parameterize the control function $u(t)$ according to the chosen subintervals, where $u(t^i)$ denotes the control

function in the i-th subinterval. In a special case, the control function can have a constant value

$$u(t^i) = u^i = x_{u^i}, \tag{22}$$

where x_{u^i} for $i = 1, \cdots, n$ are the decision variables.

- **Step 3.**
  Parameterize the initial values of both differential, as well as algebraic state trajectories

  $$z_D(t_0^i) = z_D^i = x_{z_D^i} \tag{23}$$

  and

  $$z_A(t_0^i) = z_A^i = x_{z_A^i} \tag{24}$$

  where $x_{z_D^i}$ and $x_{z_A^i}$ for $i = 1, \cdots, n$ are the decision variables.

The next step needs the following two assumptions.

Assumption 1. *The dynamics in each subinterval can be well approximated by a constant value.*

Assumption 2. *The algebraic variables in each subinterval can be well approximated by a constant value.*

- **Step 4.**
  Parameterize the left-hand side of the differential equations

  $$\dot{z}_D(t_0^i) = \dot{z}_D^i = x_{\dot{z}_D^i}, \tag{25}$$

  where $x_{\dot{z}_D^i}$ for $i = 1, \cdots, n$ are the decision variables.

- **Step 5.**
  Define the vectors of decision variables

  $$\mathbf{x}_{\dot{z}_D} = \begin{bmatrix} x_{\dot{z}_D^1} \\ x_{\dot{z}_D^2} \\ \vdots \\ x_{\dot{z}_D^n} \end{bmatrix}, \tag{26}$$

  $$\mathbf{x}_{z_D} = \begin{bmatrix} x_{z_D^1} \\ x_{z_D^2} \\ \vdots \\ x_{z_D^n} \end{bmatrix}, \tag{27}$$

  $$\mathbf{x}_{z_A} = \begin{bmatrix} x_{z_A^1} \\ x_{z_A^2} \\ \vdots \\ x_{z_A^n} \end{bmatrix}, \tag{28}$$

  $$\mathbf{x}_{u} = \begin{bmatrix} x_{u^1} \\ x_{u^2} \\ \vdots \\ x_{u^n} \end{bmatrix}, \tag{29}$$

  and finally

  $$\mathbf{X} = [\mathbf{x}_{\dot{z}_D} \quad \mathbf{x}_{z_D} \quad \mathbf{x}_{z_A} \quad \mathbf{x}_u]. \tag{30}$$

- **Step 6.**
  Transform the differential-algebraic equations into a system of the nonlinear equations. Form the simplified DAE equations for each subinterval

  $$\begin{aligned} \dot{z}_D(t^i) &= x_{\dot{z}_D^i} \\ 0 &= z_A(t^i) - x_{z_A^i} \end{aligned} \tag{31}$$

  for $i = 1, \cdots, n$, as well as the following pointwise algebraic equations representing original differential-algebraic equations

  $$\begin{aligned} x_{\dot{z}_D^i} &= F^i(x_{z_D^i}, x_{z_A^i}, x_{u^i}, p, t^i) \\ 0 &= G^i(x_{z_D^i}, x_{z_A^i}, x_{u^i}, p, t^i) \end{aligned} \tag{32}$$

  Moreover, the additional constraints to preserve continuity of the state trajectory have the following form

  $$\begin{aligned} z_D(t_F^i) &= x_{z_D^{i+1}} \\ z_A(t_F^i) &= x_{z_A^{i+1}} \end{aligned} \tag{33}$$

  for $i = 1, \cdots, n-1$.

- **Step 7.**
  Solve the nonlinear optimization problem (NLP) with the pointwise constraints using a chosen numerical optimization procedure. The NLP task can be extended by the inequality constraints imposed on the decision variables. There is also a place, where the variability constraints can be introduced in the following form

  $$x_{\dot{z}_D} < c, \tag{34}$$

where $c \in \mathcal{R}^{n_{z_D}}$.

The presented solution procedure was applied for simulation of the two-phase chemical reactor. The considered process was modeled by an index-two differential-algebraic equations.

V. The two-phase reactor optimization problem

In this Section the mathematical model of the two-phase (liquid and vapor-phase) reactor was considered. The reactants A and B are fed to the reactor as pure vapor and liquid streams with the molar flow rates F_{A0} and F_{B0}, respectively. There are two outlet streams for the liquid and vapor phases with molar flow rates F_L and F_V. Moreover, there is an assumption, that the individual phases are well-mixed and they are in physical equilibrium at appropriate pressure p and temperature T. The molar specific heat capacity c_p, density ρ, as well as latent heat of vaporization ΔH^u are constant and equal for all the species. Finally, reactant A diffuses into the liquid phase and the following reaction takes place

$$A + B \rightarrow C \tag{35}$$

The product C formation rate is given by

$$\begin{aligned} R_c &= k_o exp(-E_a/RT) C_A C_B V_L \\ &= k_o exp(-E_a/RT) N_L \rho x_A x_B, \end{aligned} \tag{36}$$

where k_o is the pre exponential factor, E_a is an activation energy, N_L is the liquid-phase molar holdup, C_A, C_B, x_A and x_B are the molar concentrations and mole fractions of the reactants A and B in the liquid phase. Moreover, V_L is the liquid holdup volume

$$V_L = \frac{N_L}{\rho}. \tag{37}$$

The considered mathematical relations depend on the observed physical phenomena. In the considered process, product C diffuses out into the vapor phase. It was assumed, that reactant B is nonvolatile. Therefore, the reactant A and the product C are present in the vapor phase. In the liquid phase all the three species are present.

The dynamic equations consist of the total mole balances in the liquid and vapor phases, the mole balance for the species A in the vapor phase, the mole balances for species A, B in the liquid phase, and the total enthalpy balance. The total enthalpy in the two phases is given by

$$H = N_L H_l + N_V H_v, \tag{38}$$

where H_l and H_v are the molar enthalpies in the liquid and vapor phases, respectively

$$H_l = c_p T \tag{39}$$

and

$$H_v = H_l + \Delta H^v = c_p T + \Delta H^v. \tag{40}$$

Moreover, the model is consisted of the algebraic relations which include phase-equilibrium relations for the species A and C present in both phases. For the phase-equilibrium relation model the Raoult's law was used

$$p y_A = P_A x_A \tag{41}$$

$$p y_C = P_C x_C \tag{42}$$

where y_A and $y_c = (1 - y_A)$ are the mole fractions of A and C in the vapor phase. P_A and P_C are the saturation vapor pressures for A and C modeled by the Antoine relation

$$P_A = exp\Big(30.5 - \frac{3919.7}{T - 34.1}\Big) \tag{43}$$

$$P_C = exp\Big(30.0 - \frac{5000}{T + 70}\Big). \tag{44}$$

Finally, the two-phase reactor was describe by the following system of differential-algebraic equations

$$\begin{aligned}
\dot{N}_V &= F_{A0} - N_A + N_C - F_V \\
\dot{y}_A &= \tfrac{F_{A0}(1-y_A)}{N_V} - \Big(\tfrac{1-y_A}{N_V}\Big) N_V - \Big(\tfrac{y_A}{N_V}\Big) N_C \\
\dot{N}_L &= F_{Bo} - F_L - R_C + N_A - N_C \\
\dot{x}_A &= -\Big(\tfrac{F_{B0} x_A + R_C(1-x_A)}{N_L}\Big) + \Big(\tfrac{1-x_A}{N_L}\Big) N_A + \Big(\tfrac{x_A}{N_L}\Big) N_C \\
\dot{x}_B &= \Big(\tfrac{F_{Bo}(1-x_B) + R_C(1-x_B)}{N_L}\Big) - \Big(\tfrac{x_B}{N_L}\Big) N_A + \Big(\tfrac{x_B}{N_L}\Big) N_C \\
\dot{T} &= \tfrac{F_{Ao}}{N_L+N_V}(T_{A0} - T) + \tfrac{F_{B0}}{N_L+N_V}(T_{B0} - T) \\
&\quad + \tfrac{R_C}{N_L+N_V}\Big(T - \tfrac{\Delta H_R}{c_p}\Big) + \tfrac{\Delta H^v}{(N_L+N_V)c_p}(N_A - N_C) \\
&\quad + \tfrac{1}{(N_L+N_v)c_p} Q \\
0 &= -x_A P_A + p y_A \\
0 &= -(1 - x_A - x_B) P_C + p(1 - y_A) \\
0 &= -N_v RT + p \tfrac{V_T \rho - N_L}{\rho}
\end{aligned} \tag{45}$$

where N_A is the molar rate of reactant A transfer from vapor to the liquid phase, N_C is the molar rate of product C transfer form liquid to the vapor phase, Q is the heat input to the reactor and V_T is the reactor volume. A description of the all process parameters and variables was presented in Table I. For this process, it is desired to control both the composition of the vapor phase y_A, as well as the temperature T. The vapor stream outlet flow rate F and the heat input Q are used as the control variables. Finally, the following differential state variables was chosen: N_V, y_A, N_L, x_A, x_B, T. The algebraic state variables are N_A, N_C, p.

The aim of the process optimization was to decrease change the set point for mole fraction of species A in vapor phase for a 15% decrease (from 0.716 to 0.600, Fig. 1). Therefore, the presented extended direct shooting approach was used. The process duration time 120 min was divided into 12 equidistance subintervals. At the beginning of each subinterval the pointwise constraints were imposed. Moreover, to preserve the continuity of the state trajectory, the additional equality constraints were introduced. The process dynamics was restricted by the pointwise variability constraints

$$0.01 \leq \dot{y}_A \leq 0.01(mol/min), \tag{46}$$

$$1.0 \leq \dot{T} \leq 1.0(K/min). \tag{47}$$

To simulate the simplified differential-algebraic equations, the *ode15 solver* was applied. Moreover, to optimize the obtained nonlinear optimization problem with pointwise-continuous constraints, the *fmincon* numerical optimization procedure was used. The obtained trajectories of the chosen differential state trajectories, as well as the control variables, were presented on the Fig. 1-4. The computations were performed with Matlab R2010a and using Intel(R) Core(TM) i7-4510 CPU 2.00 GHz 8.00 GB RAM. The computations time was equal 40 min. without parallel computing procedures.

TABLE I
THE TWO-PHASE REACTOR PARAMETERS [11]

Variable		Nominal Value
c_p	Molar heat capacity (J/mol · K)	80
E_a	Activation energy (kJ/mol)	110
F_{A0}	Inlet vapor stream molar flow rate (mol/s)	171.25
F_{B0}	Inlet liquid stream molar flow rate (mol/s)	300
F_L	Outlet liquid stream molar flow rate (mol/s)	375
F_V	Outlet vapor stream molar flow rate (mol/s)	50
k_0	Pre exponential factor (m^3/mol·s)	1.0e+11
N_L	Liquid-phase molar holdup (kmol)	12.807
N_V	Vapor-phase molar holdup (kmol)	12.839
Q	Heat input (kW)	100
R	Universal gas constant (J/mol ·K)	8.314
T	Reactor temperature (K)	341.51
T_{A0}	Inlet vapor stream temperature (K)	310
T_{B0}	Inlet liquid stream temperature (K)	298
V_T	Reactor volume (m^3)	3.0
x_A	Mole fraction of species A in liquid phase	0.238
x_B	Mole fraction of species B in liquid phase	0.677
y_A	Mole fraction of species A in vapor phase	0.716
ΔH_R	Heat of reaction (kJ/mol)	50
ΔH^v	Latent heat of vaporization (kJ/mol)	20
ρ	Liquid-phase molar density (kmol/m^3)	15

VI. CONCLUSION

In the article a new simulation approach for the industrial processes described by the index-two differential-algebraic equations. To simulate the considered DAE equations, the modified direct shooting approach was designed. The proposed procedure based on the multiple shooting method was extended by a parametrization of the left-hand side of differential equations. Therefore, in the considered task the variability constraints could be directly imposed. The original higher-index DAE equations were transformed into the simplified index-one form. The variability constraints have not ever been considered with the higher-index DAE equations simulation. The proposed methodology was investigated on the two-phase reactor simulation task.

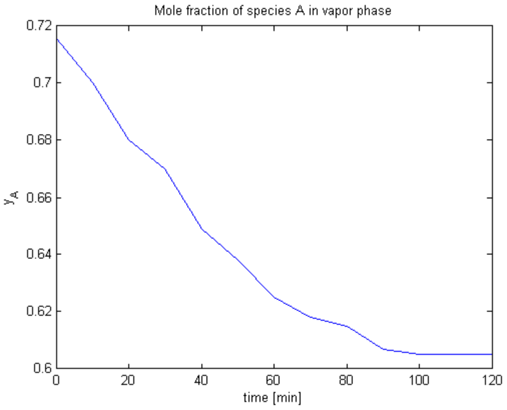


Fig. 1. Mole fraction of species A in vapor phase.

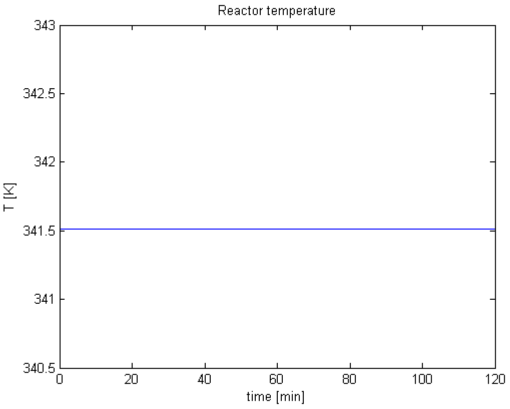


Fig. 2. Reactor temperature.

ACKNOWLEDGMENT

This work has been supported by the National Science Center under grant: DEC-2012/07/B/ST7/01216.

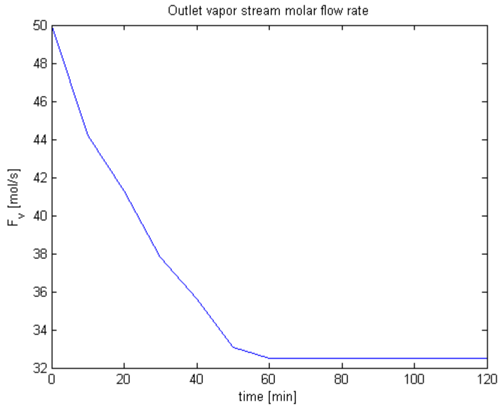


Fig. 3. Outlet vapor stream molar flow rate.

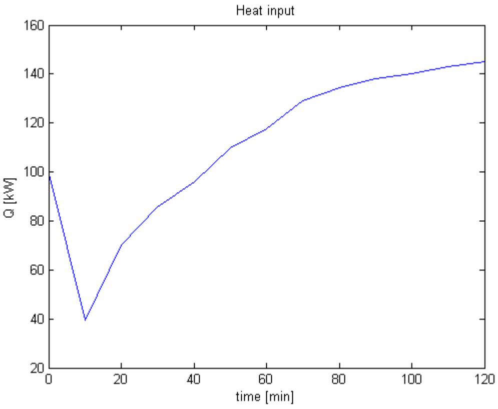


Fig. 4. Heat input.

References

[1] J.T. Betts. 2010. Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition. SIAM, Philadelphia. http://dx.doi.org/10.1137/1.9780898718577

[2] L. Biegler, S. Campbell, V. Mehrmann. 2012. DAEs, Control, and Optimization. Control and Optimization with Differential-Algebraic Constraints. SIAM, Philadelphia. http://dx.doi.org/10.1137/9781611972252

[3] J. Bonilla, F. Logist, J. Degreve, B. De Moor, J. Van Impe. 2012. A reduced order rate based model for distillation in packed columns: Dynamic simulation and the differentiation index problem. Chemical Engineering Science. 68:401-412. https://doi.org/10.1016/j.ces.2011.09.051

[4] K.E. Brenan,S.L. Campbell, L.R. Petzold. 1996. Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. SIAM, Philadelphia. http://dx.doi.org/10.1137/1.9781611971224

[5] S. Campbell, P. Kunkel. 2016. Solving higher index DAE optimal control problems. Numerical Algebra, Control and Optimization. 6:447-472. https://doi.org/10.3934/naco.2016020

[6] S. Campbell, P. Kunkel. 2009. Completions of nonlinear DAE flows based on index reduction techniques and their stabilization. Journal of Computational and Applied Mathematics. 233:1021-1034. https://doi.org/10.1016/j.cam.2009.08.111

[7] P. Daoutidis. 2015. DAEs in Model Reduction of Chemical Processes: An Overview. Surveys in Differential-Algebraic Equations II. Springer International Publishing. https://doi.org/10.1007/978-3-319-11050-9_2

[8] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, F. Allgówer. 2002. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. Journal of Process Control. 12:577-585. https://doi.org/10.1016/S0959-1524(01)00023-3

[9] P. Drąg, K. Styczeń. 2015. Multiple shooting SQP algorithm for optimal control of DAE systems with inconsistent initial conditions. Studies in Computational Intelligence. 580:53-65. https://doi.org/10.1007/978-3-319-12631-9_4

[10] J.D. Hedengren, K.V. Allsford, J. Ramlal. 2007. Moving horizon estimation and control for an industrial gas phase polymerization reactor. Proceedings of the American Control Conference 2007. 1353-1358. https://doi.org/10.1109/ACC.2007.4282820

[11] A. Kumar, P. Daoutidis. 1995. Feedback control of nonlinear differential-algebraic-equation systems. AIChE Journal 41: 619-636. https://doi.org/10.1002/aic.690410319

[12] B. Li, X. Chen, Z. Shao, J. Qian, L. Zhu, H. Li, X. Huang. 2011. Dynamic simulation and optimization of load change operation of high-temperature gas-cooled reactor pebble-bed module. PEAM 2011 - Proceedings: 2011 IEEE Power Engineering and Automation Conference. 2:381-384, https://doi.org/10.1109/PEAM.2011.6134965

[13] A. Linhart, S. Skogestad. 2009. Computational performance of aggregated distillation models. Computers and Chemical Engineering. 33:296-308. https://doi.org/10.1016/j.compchemeng.2008.09.014

[14] C. Martin-Villalba, A. Urquia, S. Dormido. 2008. Object-oriented modelling of virtual-labs for education in chemical process control. Computers and Chemical Engineering. 32:3176-3186. https://doi.org/10.1016/j.compchemeng.2008.05.011

[15] MathWorks. 2017. Documentation R2017a. https://www.mathworks.com/help/

[16] R. McKenzie, J. Pryce, N. Nedialkov, G. Tan. 2015. Regularization of nonlinear DAEs based on Structural Analysis. IFAC-PapersOnLine. 48:298-299. https://doi.org/10.1016/j.ifacol.2015.05.184

[17] K. Mohaghegh, R. Pulch, J. ter Maten. 2016. Model order reduction using singularly perturbed systems. Applied Numerical Mathematics. 103:72-87. https://doi.org/10.1016/j.apnum.2016.01.002

[18] A. Müller. 2016. A note on the motion representation and configuration update in time stepping schemes for the constrained rigid body. BIT Numerical Mathematics. 56:995-1015. https://doi.org/10.1007/s10543-015-0580-y

[19] L. Petzold. 1982. Differential/Algebraic Equations are not ODE's. SIAM Journal on Scientific and Statistical Computing,3:367-384. https://doi.org/10.1137/0903023

[20] E. Rascol, M. Meyer, J.M. Le Lann, M. Prevost. 1998. Numerical problems encountered in the simulation of reactive absorption: DAE index reduction and consistent initialisation. Computers and Chemical Engineering. Vol. 22, Issue SUPPL.1, pp S929-S932.

[21] R.R. Ratnakar, V. Balakotaiah. 2017. Bifurcation analysis of index infinity DAE parabolic models describing reactors and reacting flows. AIChE Journal. 63:295-305. https://doi.org/10.1002/aic.15568

[22] V.S. Vassiliadis, , R.W.H. Sargent, C.C. Pantelides. 1994. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. Industrial and Engineering Chemistry Research. 33:2111-2122. https://doi.org/10.1021/ie00033a014

[23] X. Wu, Y. Zeng, J. Cao. 2013. The application of the combinatorial relaxation theory on the structural index reduction of DAE. Proceedings - 12th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2013, pp. 162-166. https://doi.org/10.1109/DCABES.2013.36

[24] J. Ye, H. Xu, E. Feng, Z. Xiu. 2014. Optimization of a fed-batch bioreactor for 1,3-propanediol production using hybrid nonlinear optimal control. Journal of Process Control. 24:563-569. https://doi.org/10.1016/j.jprocont.2014.08.002

[25] V.M. Zavala, L.T. Biegler. 2006. Large-scale parameter estimation in low-density polyethylene tubular reactors. Industrial and Engineering Chemistry Research. 45:7867-7881. https://doi.org/10.1021/ie060338n

[26] V.M. Zavala, A. Flores-Tlacuahuac, E. Vivaldo-Lima. 2005. Dynamic optimization of a semi-batch reactor for polyurethane production. Chemical Engineering Science. 60:3061-3079. https://doi.org/10.1016/j.ces.2005.01.020

[27] Y. Zeng, X. Wu, J.Cao. 2014. Research and Implementation of Hungarian Method Based on the Structure Index Reduction for DAE Systems. Journal of Algorithms and Computational Technology. 8:219-231. https://doi.org/10.1260/1748-3018.8.2.219

[28] L. Zhang, D. Zhang. 2016. A two-loop procedure based on implicit Runge-Kutta method for index-3 DAE of constrained dynamic problems. Nonlinear Dynamics. 85:263-280. https://doi.org/10.1007/s11071-016-2682-8

[29] A.G. Zitlalpopoca-Soriano, E. Vivaldo-Lima, A. Flores-Tlacuahuac. 2010. Grade transition dynamic optimization of the living nitroxide-mediated radical polymerization of styrene in a tubular reactor. Macromolecular Reaction Engineering. 4:516-533. https://doi.org/10.1002/mren.200900059

Combining Cybersecurity and Cyber Defense to achieve Cyber Resilience

Darko Galinec
Department of Informatics and Computing
Zagreb University of Applied Sciences Zagreb
Zagreb, Croatia
Email: darko.galinec@tvz.hr

William Steingartner
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
Email: william.steingartner@tuke.sk

***Abstract*—Cybersecurity encompasses a broad range of practices, tools and concepts related closely to those of information and operational technology security. Cybersecurity is distinctive in its inclusion of the offensive use of information technology to attack adversaries. Use of the term ”cybersecurity” as a key challenge and a synonym for information security or IT security confuses customers and security practitioners, and obscures critical differences between these disciplines. Recommendation for security leaders is that they should use the term ”cybersecurity” to designate only security practices related to the defensive actions involving or relying upon information technology and/or operational technology environments and systems. Cyber defense is a computer network defense mechanism which includes response to actions and critical infrastructure protection and information assurance for organizations, government entities and other possible networks [3]. Within this paper, we investigate how cybersecurity and cyber defense combined may lead to cyber resilience and describe the relationships among cybersecurity, information security, operational technology (OT) security, IT security, and other related disciplines and practices e.g. cyber defense. In this regard ends, ways (processes) and means for achieving cyber resilience in today's conditions of emerging security risks are examined. Within the context of cyber resilience the novel model of cyber resilience is presented.**

***Index Terms*—cyber-attack, cyber defense, cyber resilience, cybersecurity**

I. Introduction

Cybersecurity has been practiced in military circles for over a decade. In recent years, the term has appeared in a variety of contexts, many of which have little or no relationship to the original meaning of the term. Misuse of the term obscures the significance of the practices that make cybersecurity a superset of information security, operational technology (OT) security and IT security practices related to digital assets.

The aim of this paper is to examine ways, processes and means for achieving cyber resilience in today's conditions of emerging security risks. Secondly, the aim is to create the novel model of cyber resilience that encompasses information security and cybersecurity within the context of cyber resilience (cybersecurity and emerging risks).

With the understanding of the specific environment, cyber defense analyzes the different threats possible to the given environment. It then helps in devising and driving the strategies necessary to counter the malicious attacks or threats. A wide range of different activities is involved in cyber defense for protecting the concerned entity as well as for the rapid response to a threat landscape.

These could include reducing the appeal of the environment to the possible attackers, understanding the critical locations & sensitive information, enacting preventative controls to ensure attacks would be expensive, attack detection capability and reaction and response capabilities. Cyber defense also carries out technical analysis to identify the paths and areas the attackers could target [3].

II. Basic Notions about Cybersecurity and Cyber Defence

Military terminology has migrated into nonmilitary contexts in the same fashion that military technology has migrated into civilian enterprises (e.g., the Advanced Research Projects Agency Network (ARPANET) becoming the Internet). Other terms, such as advanced persistent threat (APT; originally a euphemism for network attacks supported by the government of the People's Republic of China) [12], have endured similar transitions. In many cases, a migration of terminology is beneficial, as it develops better specificity in discussions of technology operations. However, the utility of a term is reduced when its distinctive meaning is eroded or destroyed as part of the migration to a new context.

A. Cybersecurity

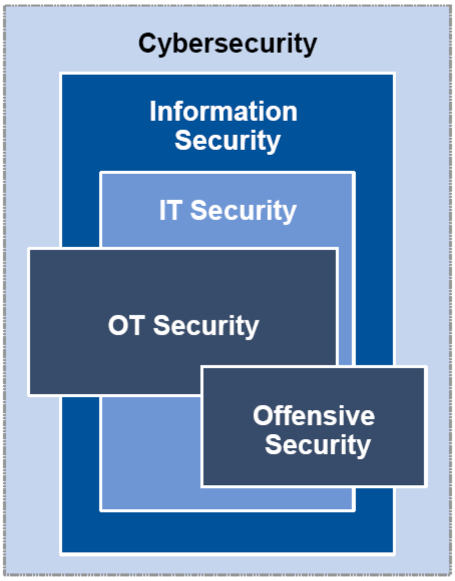
Definition: Cybersecurity is the governance, development, management and use of information security, OT security, and IT security tools and techniques for achieving regulatory compliance, defending assets and compromising the assets of adversaries [2].

According to above mentioned authors, cybersecurity:

- is a superset of the practices embodied in IT security, information security, OT security and offensive security (see Figure 1);
- uses the tools and techniques of IT security, OT security and information security to minimize vulnerabilities, maintain system integrity, allow access only to approved users and defend assets;
- includes the development and use of offensive IT- or OT-based attacks against adversaries;

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

- supports information assurance objectives within a digital context but does not extend to analog media security (for example, paper documents).



Source: Gartner (June 2013)

Fig. 1. Components of Cybersecurity

But, in the same time, cybersecurity is not:

- merely a synonym for information security, OT security or IT security;
- use of information security to defend an enterprise against crime;
- Cyberwarfare – although the definition of this term is still controversial, the consensus is that ”cyberwarfare” refers to the use of cybersecurity capabilities in a warfare context. This is a complex area and should not be confused with physical attacks against infrastructure (e.g. destruction of property and machinery) and information warfare (e. g. applying psychological operations through propaganda and misinformation techniques).
- Cyberterrorism – In a similar fashion to cyberwarfare, ”cyberterrorism” refers to the use of cybersecurity techniques as part of a terrorist campaign or activity.
- Cybercrime – Cybercrime is merely an affected or pretentious term for criminal attacks using IT infrastructure. It is not related to cybersecurity.

Appropriate uses of ”cybersecurity” [12] would be:

- in response to threat risk assessments, the department increased its cybersecurity investment to enable reductions in vulnerabilities and increased capabilities for counterattacks against identified attackers (integration of IT security and offensive capabilities in a single program).
- Integration of the IT and OT security programs within the cybersecurity team enables more holistic responses to threats (integration of IT and OT in a single program).
- The ”hacktivist” organization Anonymous employs a variety of cybersecurity techniques to forward its agenda (use of offensive capabilities).

However, we could face with some inappropriate uses of ”cybersecurity”:

- In order to mitigate the theft of laptops, the store’s cybersecurity plan calls for the use of whole drive encryption. (This describes a basic IT security action.)
- The cybersecurity policy mandates the use of complex passwords for all CAM systems on the factory floor. (This describes a basic OT security requirement.)

B. Cyber Defense

There are no common definitions for Cyber terms - they are understood to mean different things by different nations/organizations, despite prevalence in mainstream media and in national and international organizational statements [9].

However, [3] gives definition and further explanation of term cyber defense as follows: Cyber defense is a computer network defense mechanism which includes response to actions and critical infrastructure protection and information assurance for organizations, government entities and other possible networks.

Cyber defense focuses on preventing, detecting and providing timely responses to attacks or threats so that no infrastructure or information is tampered with. With the growth in volume as well as complexity of cyber-attacks, cyber defense is essential for most entities in order to protect sensitive information as well as to safeguard assets.

Cyber defense provides the much-needed assurance to run the processes and activities, free from worries about threats. It helps in enhancing the security strategy utilizations and resources in the most effective fashion. Cyber defense also helps in improving the effectiveness of the security resources and security expenses, especially in critical locations.

By the recognition of the need to accelerate detection and response to malicious network actors, the United States (US) Department of Defense (DoD) has defined a new concept, Active Cyber Defense (ACD) as DoD’s synchronized, real-time capability to discover, detect, analyze, and mitigate threats and vulnerabilities [14].

III. Cybersecurity Starategy and Risk Management

While the cost of defending cyber structures as well as the payoffs from successful attacks keeps rising, the cost of launching an attack simultaneously keeps decreasing [7].

By the standard military definition, ”strategy” is the utilization of all of a nation’s forces, through large-scale, long-range planning and development, to ensure security or victory. For traditional wars against traditional monolithic opponents, that approach worked.

However, for today’s world of asymmetric warfare and rapidly changing threats, the medical definition of strategy from Merriam-Webster’s dictionary is more appropriate for addressing cybersecurity: ”an adaptation or complex of adaptations (as of behavior, metabolism or structure) that serves or appears to serve an important function in achieving evolutionary success."

The key to increasing cybersecurity is getting to lower levels of vulnerability. Although threat awareness is important, by reducing vulnerabilities, all attacks are made more difficult [10].

A. Cybersecurity Risk Management

Cyber security breaches, such as those at Ashley Madison, the US Office of Personnel Management and JP Morgan Chase have demonstrated the real and present threat from cyber breaches. Director of the National Security Agency and head of the United States Cyber Command, Admiral Mike Rodgers has been moved to state that "It's not about if you will be penetrated but when" [8].

If there isn't sufficient visibility of cyber security status, organizations won't be able to manage cyber security risks and they will almost certainly suffer a breach. "Visibility of cyber security status" means having the complete picture, with measurements so that we can answer the following questions:

- What are our current measured levels of cyber security risk across the Enterprise from the multiple threats that we face?
- Are these cyber security risks tolerable?
- If not, what is our justified and prioritized plan for managing these risks down to tolerable levels?
- Who is responsible and by when?

The ability to measure cyber security status is fundamental; if we can't measure then we can't manage. Security incident and event management (SIEM) and data analytics solutions can provide valuable indications of actual or potential compromise on the network but these are partial views, indicators of our overall risk status but not measurements of our risk status.

Similarly, threat intelligence services can identify data losses and provide valuable indications of actual or impending attacks but again these are not measurements of our risk status. The same can be said individually about outputs from compliance management, vulnerability management, penetration testing and audits.

Only by pulling together all of the relevant indicators and partial views we can develop overall risk-based measurement and visibility of our cyber security status [8]. When confidence in our cybersecurity risk measurements exists it is possible to respond to events and make decisions quickly, e. g.:

- Be able to identify risks that we aren't prepared to tolerate and have a clear and prioritized risk-based action plan for the control improvements necessary to reduce these risks to an acceptable level
- To have a better understanding of the implications from threat intelligence or outputs from SIEM and data analytics allowing faster, better targeted responses
- To develop risk-based justifications for investment in cyber security solutions and services.

But with the very high level of threat and high rates of change in both the threat and control landscapes we need to be able to refresh our view of our cyber security status on an almost daily basis.

Cybersecurity risk management which previously might have been an annual process as part of planning and budgeting is now a critical real-time facilitator in the battle against cyber breaches [8]. Cyber security breaches occur when people, processes, technology or other components of the cyber security risk management system are missing, inadequate or fail in some way. So we need to understand all of the important components and how they inter-relate.

This doesn't mean that risk management system needs to hold details of (for example) every end point and the status of every vulnerability on the network because there are other tools which will do that but the risk management system does need to know that all end points on the network have been (and are being) identified and that critical vulnerabilities are being addressed quickly.

Cybersecurity success is essentially the result of an effective risk management process. However, this process is being challenged by the inherent complexity of systems, developed with vulnerable components and protocols, and the crescent sophistication of attackers, now backed by well-resourced criminal organizations and nations.

B. Cyber Resilience

With this scenario of uncertainties and high volume of events, it is essential the ability of cyber resilience.

Cyber resilience is the ability of a system, organization, mission, or business process to anticipate, withstand, recover from, and adapt capabilities in the face of adversary conditions, stresses, or attacks on the cyber resources it needs to function.

Cyber resilience from an organizational perspective is defined as "the ability to continuously deliver the intended outcome despite adverse cyber events", and this definition is systematically described and justified [1].

Starting with the 2012 World Economic Forum meeting in Davos, cyber resilience [1] has been not only an area of growing importance for individuals, businesses and societies, but also a concept that has gained in attention and usage.

Cyber resilience refers to the ability to continuously deliver the intended outcome despite adverse cyber events. The notion of continuously, means that the ability to deliver the intended outcome should be working even when regular delivery mechanisms have failed, during a crisis and after a security breach. The notion also denotes the ability to restore the regular delivery mechanisms after such events as well as the ability to continuously change or modify these delivery mechanisms if needed in the face of changing risks. The intended outcome refers to that which the unit-of-analysis (e.g. the nation, organization or IT system) is intended to achieve, such as the goals of a business or business process or the services delivered by an online service [1].

IV. Cyber Resilience Context

Cybersecurity is an inherently distributed problem that will continue to evolve at the speed of technology. According to the

11th Annual Global Information Security Survey, conducted by PriceWaterhouseCoopers and CSO Online [6], executives remain confident in the robustness of their security initiatives. In the survey, 84% of CEOs and 82% of CIOs contend their cyber security programs are effective, while 78% of chief information security officers express full confidence in their existing cyber security programs. With breaches on the rise, companies should focus on cyber resilience, not just cyber security. The number of security incidents detected is rising significantly year-over-year climbing from 2,989 reported in 2012 to 3,741 in 2013. Add to that the fact that the average losses per incident are up 23% year-over-year, and that the number of organizations reporting losses of more than $10 million per incident is up 75% from just two years ago [5].

Cyber security isn't going far enough so Cyber Resilience must be taken into consideration. Once businesses accept that cyber attacks will be made against their organizations and will be successful, they can move to the next step: implementing a Cyber Resilience Program (CRP). A CRP encompasses the ideas of defense and prevention, but goes beyond those measures to emphasize response and resilience in moments of crisis [5].

A. Emerging Risks in Cybersecurity

Today's security professionals battle threats from outside the organization as well as those from their own employees. But what about threats that they already know exist? The next few years will see a variety of attacks as well as progress in the technologies and processes that prevent them. Gartner's predictions focus on how organizations can prepare for future cybersecurity risk while taking appropriate action today.

1) Through 2020, 99% of vulnerabilities exploited will continue to be ones known by security and IT professionals for at least one year. Recommended Action: Companies should focus on fixing the vulnerabilities they know exist. While these vulnerabilities are easy to ignore, they're also easier and more inexpensive to fix than to mitigate.
2) By 2020, a third of successful attacks experienced by enterprises will be on their shadow IT resources. Recommended Action: Business units deal with the reality of the enterprise and will engage with any tool that helps them do the job. Companies should find a way to track shadow IT, and create a culture of acceptance and protection versus detection and punishment.
3) By 2018, the need to prevent data breaches from public clouds will drive 20% of organizations to develop data security governance programs. Recommended Action: Develop an enterprise-wide data security governance (DSG) program. Identify data security policy gaps, develop a roadmap to address the issues and seek cyber insurance when appropriate.
4) By 2020, 40% of enterprises engaged in DevOps will secure developed applications by adopting application security self-testing, self-diagnosing and self-protection technologies. Recommended Action: Adopt Runtime application self protection (RASP) for DevOps. Evaluate less mature vendors and providers for potential security options.
5) By 2020, 80% of new deals for Cloud Access Security Broker (CASB) technology will be packaged with network firewall, secure web gateway (SWG) and web application firewall (WAF) platforms. Recommended Action: While concerns exist about customer migration to the cloud and bundling purchases, companies should assess the application deployment roadmap and decide whether investment is justified.
6) By 2018, enterprises that leverage native mobile containment rather than third-party options will rise from 20% to 60%. Recommended Action: Experiment and become familiar with native containment solutions. Keep in mind that enterprise with average security requirements should plan to move gradually to native containment.
7) By 2019, 40% of IDaaS implementations will replace on-premises IAM implementations, up from 10% today. Recommended Action: Enough limitations have disappeared on Identity as a Service (IDaaS) that companies should start experimenting on small-scale projects. While a clash of regulations could derail the increased implementation, companies should work to recognize the current limitations and benefits.
8) By 2019, use of passwords and tokens in medium-risk use cases will drop 55%, due to the introduction of recognition technologies. Recommended Action: Passwords are too entrenched in business practices to disappear completely, but companies should look for products that focus on development of an environment of continuous trust with good user experience. Begin by identifying use cases, and press vendors for biometric and analytic capabilities.
9) Through 2018, over 50% of IoT device manufacturers will not be able to address threats from weak authentication practices. Recommended Action: By changing the enterprise architecture, IoT introduces new threats. Early IoT security failures might force the industry towards authentication standards, but companies should identify authentication risks, establish identity assurance requirements, and employ metrics.
10) By 2020, more than 25% of identified enterprise attacks will involve IoT, though IoT will account for only 10% of IT security budgets. Recommended Action: As IoT continues to grow, vendors will favor usability over security and IT security practitioners remain unsure of the correct amount of acceptable risk. Companies should assign business ownership of IoT security, focus on vulnerable or unpatchable IoT devices, and increase IoT-focused budget [13].

B. Information Security, Cybersecurity and Cyber Resilience

Cybersecurity is no longer enough: there is a need for strategy of defense, prevention and response. The idea of resilience, in its most basic form, is an evaluation of what happens before, during and after a digitally networked system encounters a threat. Resilience should not be taken to be synonymous with ”recovery”. It is not event-specific: it accrues over the long term and should be included in overall business or organizational strategy. Resilience in context of ability of systems and organizations to withstand cyber events means the preparations that an organization has made with regard to threats and vulnerabilities, the defenses that have been developed, and the resources available for mitigating a security failure after it happens. Normalization is the key. Cyber risk should be viewed just like any other risk that an organization must contend with in order to fulfil its goals. Leaders of business and government need to think about resilience for two reasons: first, by doing so they avoid the catastrophic failure threatened by an all-or-nothing approach to cyber risks (i.e. preventing network entry as the only plan), and second, it ensures that the conversation goes beyond information technology or information security [2].

The first point, that a long-term view and durability are key factors in ensuring cyber resilience, does not need further explanation. A plan that encompasses actions and outcomes before, during and after the emergence of a threat will generally be superior to a plan that only considers one instance in time. The second point, that leaders must broaden the conversation, merits more attention. It is vital to our economic and societal resilience that we think beyond information security to overall network resilience that ensures we can deal with existing risks and face new risks that will come with such things as artificial intelligence, the internet of things or quantum computing. In order to ensure long-term cyber resilience, organizations must include in their strategic planning the ability to iterate based on evolving threats from rapidly evolving disruptive technologies [2].

By promoting an overall cyber-resilience approach, long-term strategy (including which technologies a business will implement over the next five, 10 or more years) is a continual strategic conversation involving both technology and strategic leaders within an organization. The cyber-resilience approach ensures greater readiness and less repetition making it, on the whole, more efficient and more effective. Security, in contrast to resilience, can be seen as binary. Either something is secure or it isn’t. It is often relegated to a single, limited technical function, keeping unauthorized users out of a networked system [2].

While there are many broader definitions of cybersecurity, there is a difference between the access control of cybersecurity and the more strategic, long-term thinking cyber resilience should evoke. Additionally, since vulnerability in one area can compromise the entire network, resilience requires a conversation focused on systems rather than individual organizations. For networked technologies, vulnerability in one node can affect the security and resilience of the entire network. Therefore, resilience is best considered in the context of a public good or ”commons”. That’s why partnerships are keys. These can be between businesses as well as with regulators, prosecutors and policy-makers [2]. Since cyber resilience is really a matter of risk management, there isn’t a single point at which it begins or ends. Instead, it comes from building strategy and working to ensure that the risk-transfer mechanisms that work for more traditional threats are also brought to bear on new cyber threats. Responsibility for cyber resilience is question of strategy rather than tactics. Being resilient requires those at the highest levels of a company, organization or government to recognize the importance of avoiding and mitigating risks. While it is everyone’s responsibility to cooperate in order to ensure greater cyber resilience, leaders who set the strategy for an organization are ultimately responsible, and have increasingly been held accountable for including cyber resilience in organizational strategy [2]. The real cybersecurity challenge is the unknown. Former US Secretary of Defense Donald Rumsfeld gave the explanation of this during a news briefing in 2002: ”There are known knowns. These are the things that we know. There are known unknowns. That is to say, there are things that we know we don’t know. But there are also unknown unknowns. These are things we don’t know we don’t know [11].

Combating known threats is an essential part of a cybersecurity strategy. It goes alongside advanced capabilities to anticipate, capture and ultimately learn from unknown threats. Systems have different weak spots and different processes (challenges) and they each manage risk in different ways (solutions). In other words, to each security challenge (evaluated as ”known” or ”unknown”) corresponding solution to that challenge exists (evaluated as ”knowns” or ”unknowns”). By incorporating values obtained during the system security assessment process into the model we get ”known knowns” relating to information security, ”known unknowns” relating to cyber security and ”unknown unknowns” related to cyber resilience [4].

Example: There is a known crisis in the cybersecurity workforce: a massive shortfall in qualified and trained security professionals. There is also an unknown solution to this crisis. The broad and growing scope of the challenge requires a corresponding broadening of skill sets that are both known and unknown [11].

Finally, Cyber Resilience Model structure and content is presented (Figure 2), consisting of information security (CIA triad threats and responses to them i.e. - known knowns), cybersecurity (non-CIA complex threats, APTs and corresponding responses to them i.e. known unknowns) and cyber resilience (unforeseeable and unpredictable threats and responses to them unknown unknowns).

There are opportunities around those cybersecurity solutions that can take the fear factor out of unknown quantities, and make them ”known”. But there continue to be significant opportunities around those protection measures that apply the universe of known cyber threat knowledge, to keep the system continuously secure [4].

In order to cope with the growing challenges, which today are manifested as unknown unknowns, systems tend to enable personnel and adjust existing and develop new processes, organization and technology. Technologies are being developed

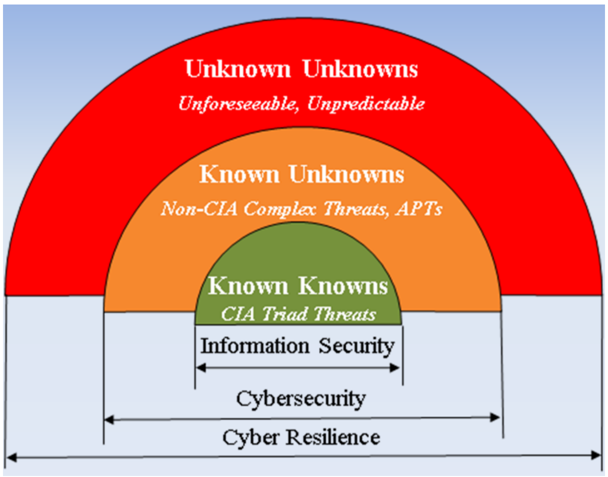


Fig. 2. Cyber Resilience Model

which, unlike traditional approaches, have the ability to protect system from serious threats by learning what is ”normal” for the organization and its people and thereby spotting emerging anomalies. Unlike, the traditional rules and signature based approach, the technology can spot threats that could harm organization and network that the traditional approaches are unable to detect. It can deal with uncertainty and delivers adaptive protection for organizations from both insider threats and advanced cyber-attacks.

V. Conclusion

Nowhere has technological development been more dynamic and comprehensive than in the area of communication and information technology. The focus has always been on the rapid development and introduction of new services and products, while the security-related aspects usually had little influence on the broad acceptance of new technologies.

The life cycles of modern-day information systems, from the process of planning, introduction and usage to their withdrawal from use are very short, which often makes their systematic testing impossible and is most commonly applied as an exception, in expressly prescribed cases.

Modern societies are deeply imbued with communication and information technology. People are nowadays connected using various technologies for the transmission of text, image and sound, including the increasing Internet of Things (IoT) trend. Deviations in the proper operation of these interconnected systems or their parts are no longer merely technical difficulties; they pose a danger with a global security impact. Modern societies counter them with a range of activities and measures collectively called cybersecurity.

In our paper the ways, processes and means for achieving cyber resilience in today’s conditions of emerging security risks are examined. Within the context of cyber resilience (cybersecurity and emerging risks) the novel model of cyber resilience that encompasses information security and cybersecurity is presented. Further investigations of ours are directed towards finding and enabling efficient and effective processes for agile (adaptable, aware, flexible and productive) cyber resilience of the security information system able to cope with unforeseeable and unpredictable events (unknown unknowns) in inner and outer environment of the system as a whole. Key roles related to that goal have people (actors) and their performance at all levels of systems hierarchy (cybersecurity combined with cyber defence).

Acknowledgment

This work has been supported by Grant No. 002TUKE-4/2017: Innovative didactic methods of education process at university and their importance in increasing education mastership of teachers and development of students’ competences.

References

[1] Björck F. et al.: Cyber Resilience Fundamentals for a Definition. In: Rocha A., Correia A., Costanzo S., Reis L. (eds) New Contributions in Information Systems and Technologies. Advances in Intelligent Systems and Computing, vol 353. Springer, Cham, 2015.

[2] Dobrygowski, D.: Cyber resilience: everything you (really) need to know, available at https://www.weforum.org/agenda/2016/07/cyber-resilience-what-to-know/, Accessed: 21st June 2017.

[3] Cyber Defense, available at https://www.techopedia.com/definition/6705/cyber-defense, Accessed: 10th February 2017.

[4] Exclusive Networks: Unknown Unknowns The Ultimate Test for Cybersecurity, available at http://www.exclusive-networks.com/uk/blog/unknown-unknowns-ultimate-test-cybersecurity/, Accessed: 1st June 2017.

[5] Goche, M., Gouveia, W.: Why Cyber Security Is Not Enough: You Need Cyber Resilience, available at https://www.forbes.com/sites/sungardas/2014/01/15/why-cyber-security-is-not-enough-you-need-cyber-resilience/#562402a21bc4, Accessed: 1st June 2017.

[6] Hulme, G.V.: Security spending continues to run a step behind the threats, available at http://www.csoonline.com/article/2134074/strategic-planning-erm/security-spending-continues-to-run-a-step-behind-the-threats.html, Accessed: 3rd June 2017.

[7] Infosecurity, available at http://infosecurityinc.net/wp-content/uploads/2011/07/Consult-Cyber-1Cyber-Threats-Diminishing-Attack-Costs-gaIncreasing-Complexity4.jpg, Accessed: 15th November 2016.

[8] Marvell, S.: The real and present threat of a cyber breach demands real-time risk management, Acuity Risk Management, 2015.

[9] NATO Cyber Cooperative Cyber Defence Center of Excellence Tallin Estonia, available at https://ccdcoe.org/cyber-definitions.html, Accessed: 10th February 2017.

[10] Pescatore, J.: Toward a National Cybersecurity Strategy, G00167598, Gartner, Inc., 2009.

[11] Tucker, E.: Official: FBI probing attempted cyber breach of NY Times, available at http://www.federaltimes.com/articles/official-fbi-probing-attempted-cyber-breach-of-ny-times, Accessed: 31st May 2017.

[12] Walls, A., Perkins, E., Weiss, J.: Definition: ”Cybersecurity”, G00252816, Gartner, Inc., 2013.

[13] Wheeler, J.A.: Emerging Risks in Cybersecurity: Gartner’s Top Ten Predicitons, available at http://blogs.gartner.com/john-wheeler/gartner-top-ten-cybersecurity-predicts/, Accessed: 2nd June 2017.

[14] United States Department of Defense: Strategy for Operating in Cyberspace, Department of Defense, 2011.

Zusammenfassung

Die Internetsicherheit umfasst viele Verfahren, Testinstrumente und Begriffe, die in einem engen Zusammenhang mit der Informations-und Operationssicherheit in Technologien sind. Das Risiko der Internetsicherheit besteht darin, dass sie auch offensive Benutzung der Informationstechnologien zu den Angriffen umfasst. Die Benutzung des Begriffs Internetsicherheit als eine Hauptherausforderung und das Synonym für die Informationssicherheit oder IT-Sicherheit verwirrt die Kunden und die IT-Spezialisten und verheimlicht kritische Unterschiede zwischen diesen Disziplinen. Man empfiehlt den Experten im Bereich der Sicherheit, den Begriff ”Cybersecurity” zu benutzen, damit sie nur die Sicherheitsverfahren bestimmen. Diese sind in einem engen Zusammenhang mit den Abwehrprozessen. Diese Abwehrprozesse umfassen oder verlassen sich auf die Informationstechnologien oder auf die Systeme und die Umfelder der Operationstechnologien. Der Internetabwehr ist der Abwehrmechanismus des Computernetzes, der die Reaktion auf Angriffe umfasst, Schutz gegen eine gefährliche Netzinfrastruktur und Informationsschutz für Organisationen, Regierungsorganisationen und andere mögliche Netze. Im Rahmen dieses Artikels forschen wir, wie die Internetsicherheit in der Verbindung mit den Angriffen in Internet zu der Widerstandsfähigkeit führt. Zugleich beschreiben wir die Beziehungen zwischen der Internetsicherheit, Informationssicherheit und anderen verwandten Disziplinen und Verfahren, wie z. B. Angriffe in Internet. In dieser Hinsicht werden die Ziele, Verfahren und Testinstrumente geforscht, die zum Erzielen von Cyber-Widerstandsfähigkeit in heutigen Bedingungen der zunehmenden Sicherheitsrisiken dienen. Im Kontext der Cyber-Widerstandsfähigkeit wird ein neuer Model der Cyber-Widerstandsfähigkeit präsentiert.

WLAN heat mapping in hybrid network

Kornél Gyöngyösi*, Péter János Varga* and Zsolt Illési**
*Kandó Kálmán Faculty of Electrical Engineering,
Óbuda University, Budapest, Hungary
** National College of Ireland, Ireland
Email: kornel.gy@gmail.com, varga.peter@kvk.uni-obuda.hu, zsolt@illesi.hu

Abstract— Wireless Local Area Network (WLAN) have become popular because of mobility and easy installation. Nowadays, WLAN access points (APs) are everywhere. For proper operation of own hybrid network, it may be necessary to know the APs locations. The visualization can help to get the best coverage or the best conditions of the network. In addition, the measurements help to create an AP inventory or identify unauthorized and illegal APs. Measurements can be divided into two large groups, outdoor and indoor measurements. The hardware and software environment of the two procedures may be different. The goal of this article is to present reproducible measurement to WLAN APs visualization before or after hybrid network installations.

***Keywords*–WLAN, AP, visualization, localization**

I. Introduction

Wireless network penetration is continuously increasing. Since 2012 we carry out measurements in Budapest on a 29 km long route to be able to determine the ratio and the nature of the expansion. In the route there are garden suburbs, industrial areas, multi story block of flats and offices. During our measurements, we aimed to use up to date tools and software. The hardware measurement apparatus was always an outdoor USB WLAN receiver and GPS tracker. The measurement software kept changing, depended on the operating systems and the hardware support. Our scans revealed the rising tendency of using wireless devices. To be able to get a comprehensive view in relation to the correctness of the collected data, we compared it to an international WLAN device database which is provided by the WIGLE (Wireless Geographic Logging Engine) webpage free for everybody. You can see the measured data and WIGLE data shown in Figure 1. [1], [2].

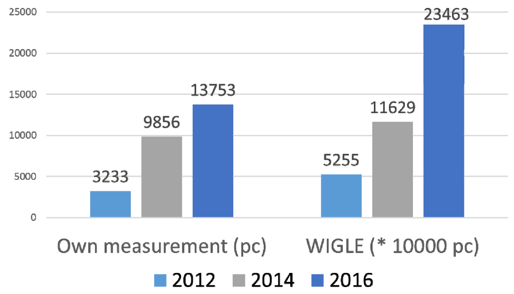


Figure 1. WLAN measurement data and WIGLE data comparison

The measured and stored data–besides the number of the identified WLAN devices–includes the following data about the networks:

- SSID (Service Set Identifier),
- MAC Address,
- RSSI (Received signal strength indication),
- Channel and Channel Width
- ,802.11 standard,
- Security and authentication mechanisms,
- GPS coordinates.

During the data evaluation, it was necessary to provide visualization of the identified WLAN devices, especially the APs. We were looking for a geographical information system, which is able to filter the recorded data on the basis of different parameters. To accomplish this, we were looking for a solution, which is able import and display the collected data in a geographical information system. It is possible to convert data with different utilities, but using a specialized website to fulfil these tasks makes the procedure way easier. For our tests, we selected GPS Visualizer page as the adequate conversion platform.

During the conversion, some data may be lost (e.g. date, signal strength, channel number etc.), therefore it is practical to check different options to ensure the comprehensiveness of both the analysis and visualization. There are specialized websites, which provide such functionality. However, these are able to display only a subset of the collected information.

Google Maps or Google Earth portals for example provide such services. Using the converted log file the webpage is able to fit and plot the points on the map. This is shown in Figure 2. [3].

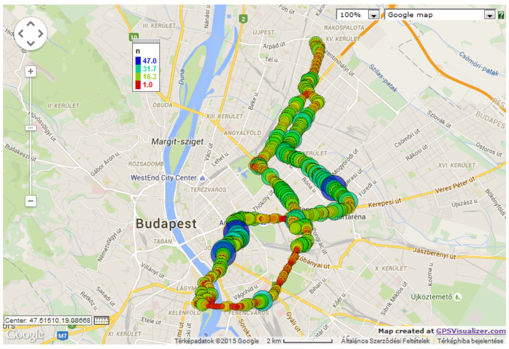


Figure 2. Measured data visualization on Google Maps with position information

On this picture, it is apparent that the number and dispersion of the devices are high in Budapest city center and in the block of flats.

At the beginning of our WLAN test series, the question of the potential methods for AP radio-signal visualization also was raised. Finally, we selected the Footprint application, which is able to filter and group the measured signal strength based on the MAC address. By selecting a MAC address, the application separates out the measurement data and based on this information generates two images. One image shows the route of the measurer, and the other one represents the radio heat map of the AP. The application associates different colors to signal strength values as shown in Table 1.

TABLE I.
WLAN RSSI AND COLORS

RSSI (dB)	Colors
-100dB < RSSI < -95dB	transparent -> blue
-95dB < RSSI < -90dB	blue -> purple
-90dB < RSSI < -80dB	purple -> red
-80dB < RSSI < -70dB	red -> yellow
-70dB < RSSI < -60dB	yellow -> green
-70dB < RSSI < -20dB	green

Based on the calculated coordinates and the two images that generated by Footprint, Google Earth provide a combined visualization, which identifies the exact location of the AP and the signal strength of the WLAN network. The combined heat map is shown in Figure 3.

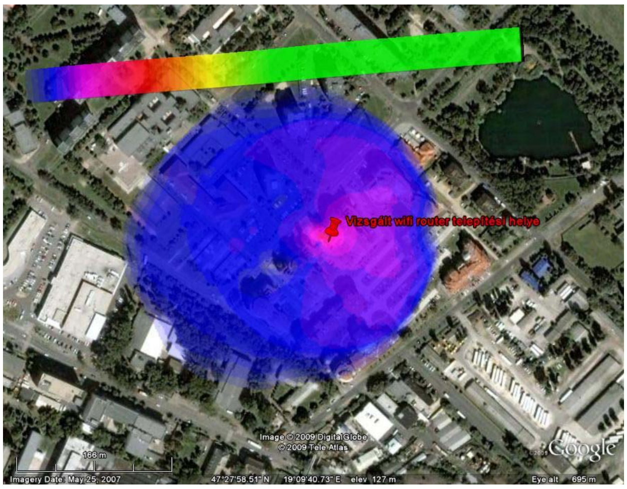

Vizsgált wifi router telepítési helye
Image © 2009 DigitalGlobe
© 2009 Tele Atlas
166 m
©2009 Google
Imagery Date: May 25, 2007
47°27'58.51" N
19°09'40.73" E
elev 127 m
Eye alt 695 m

Figure 3. WLAN signal heat map in Google Earth

The map-based illustration enables to visually identify our AP's and the neighboring AP's network coverage. Identifying all APs, generating the related geographical and signal strength data, and display them on a single surface, but distributing them on different layers facilitates the comprehensive analysis. The visual analysis of our own network may require the comparative analysis with all the neighboring networks. This is a more complex task, which requires both specialized hardware and software tools.

II. INTRODUCTION

The proliferation of WLAN devices changed our daily life and customs. You may find WLAN infrastructures not only in the homes, but in the workplaces and the public squares of the cities. Our measures support this conclusion. The high number and independent deployment of wireless networks, however, is a possible source of error for both the network operators and users. As long as we are operating our own local WLAN in our networking environment, managing potential operational errors is easier. However, if there are multiple, independently deployed and operated systems around our AP, this generates radio frequency problems for our networking devices. This issue reduces the access capacity of our network. Communication becomes indeterminate, which in worst case may cause complete network connection unavailability. This is why adequate assessment and planning are necessary before developing a new network.

A. Document User Requirements

The first step of deploying a WLAN system is defining user requirements, which provides information and analysis about the environment and usage of the new network. User requirement shall include the expected indoor and outdoor network coverage, the minimum data transfer rate, network capacity and any relevant system characteristics of the applications, and business functions. Before installing the network, it is necessary–bearing in mind the optimal utilization–to discover the radio frequency characteristics of the construction environment.

B. WLAN Analysis

The second step is to measure the emanated signals of the radio systems at the site, which facilitates the system design, deployment and operations. It is necessary to measure it at the planned place of deployment–both indoor and outdoor.

We used the following hardware tools to measure the radio frequency:

- laptop Windows with Windows OS,
- TP-Link Archer T4UH AC1200 Wireless Dual Band USB adapter (802.11 a/b/g/n/ac),
- AirView Spectrum Analyzer (2,4GHz),
- USB GPS/GLONASS receiver.

The software measurement environment consists of:

- Acrylic Wi-Fi Professional,
- Acrylic Wi-Fi HeatMaps,
- Ubiquiti AirView.

The designer gets a comprehensive picture after reviewing the data collected and analyzed by the measuring system about:

- the radio frequency saturation and utilization of the deployment location and its environment,
- the radio frequency interference and jamming properties of the location to be covered.

The beginning of the measurement shall commence with an indoor and outdoor monitoring, which is made with the Acrylic Wi-Fi Professional application, that is installed to the measurement laptop. It is possible to decide with this application the:

- 802.11 Version: Detect WLAN APs and clients type;
- Information on maximum data transmission rate supported by access points;
- Statistics on packets retried by APs and WLAN devices;
- Performance and behaviour details on all WLAN devices in range.

These pieces of information are the part of the overall analysis. It is necessary to investigate the radio frequency contamination of the environment both on 2.4GHz and 5GHz. Based on our preliminary wireless network distribution survey, it is known that the number of 5GHz is insignificant, therefore a thorough checking of the 2.4GHz band is suggested. This measurement can be made with a USB spectrum analyzer, which provides a visual WLAN radio channel analysis. The result of the measurement is shown in Figure 4. On the top of the figure, the reserved 2.4GHz channels are displayed. In the middle, the current traffic on a radio segment is indicated. At the bottom, the recorded radio top values are shown[4].

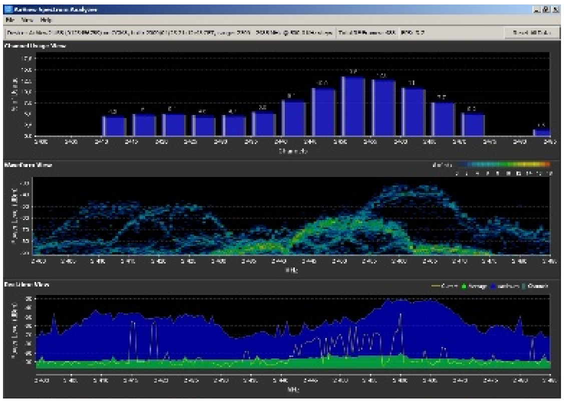


Figure 4. 2.4 GHz spectrum radio frequency analysis

C. Network Design

The two types of measurements, detailed above, provide support to design a new network. The new WLAN network–based on the documented user requirements and the reconnaissance–is designed with a computer program. The aim of the design is to optimally deploy the APs, e.g. the physical nodes of the network on the premises. This will guarantee the resilience of the system. The location of the APs is the function of physical properties of the deployment site, the specifications of the aerials and the user requirements. The design criteria for the design is compound, the designer needs to consider

- the structural materials of the building,
- the size and location exposure to radiation,
- the blind sectors,
- the electromagnetic interference minimization of the environment [5].

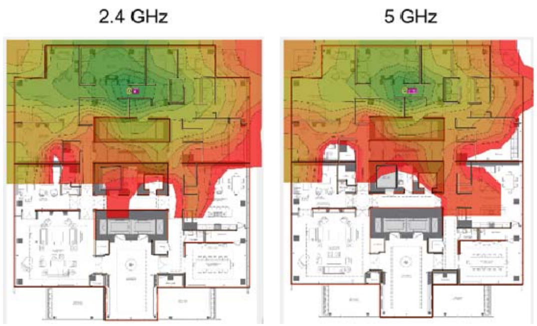


Figure 5. Visual simulation of a 2.4GHz and a 5 GHz APs on the layout of the same office

In this phase of the network design, the designer shall detail the network components of both the wireless and the wired network. At this point in time, the wired and the wireless solutions link. To finalize the comprehensive network layout, the designer shall create the hybrid network topology.

D. Network Deployment

It is advised to have a final physical walk-trough and analysis before installing the devices to their final locations. This shall cover the temporary installation of the APs, and the follow up measurements. It is necessary to check the factual radiation of the APs and to be able to guarantee the anticipated network services. After this step, comes the final installation of the wireless devices and linking these to the wired LAN. During the network installation, each individual active and passive component shall be tested and the test result shall be documented.

E. Post installation Steps

After the installation, when the furniture and other equipment are in their final position, it is suggested to take final control measures. This could confirm the operability of the system.

A laptop and a measurement application is needed to accomplish this test. In this case this application is Acrylic Wi-Fi HeatMaps. In the application, it is necessary to enter the design layout and the pre-defined measurement points. The number of measurement points depends on the design precision. The suggested quantity different for indoor and outdoor deployment sites. In case of an outdoor network installation, when it is possible identify the APs with high precision, the number of measurement points can be reduced significantly. In case of an indoor network installation, it is practical to increase the number of measurement points to increase precision. It is essential to precisely measure AP location in a coordinate-correct display in the software for outdoor installation. The USB GPS/GLONASS receiver for these measurement is used. The tool identifies the measurement points on a self-defined measurement route. Therefore, the design layout of the deployment location must be length calibrated in the measurement software. The software displays graphically all determined radio frequency measurement data on a proportionally scaled layout diagram. It ensures the comprehensive analysis of the measured data. The software not only collects measurement data, but it also generates the preferred heat map, based on the selection criteria. The software, besides visualization, also produce the test report of the measurements. Figure 6. shows the heat map of an office building [6].

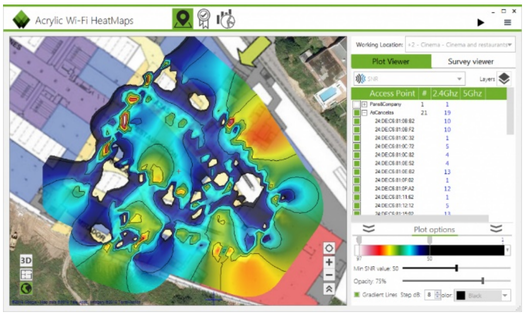


Figure 6. Heat map of a complex WLAN

Test data, and analysis can highlight misconfigured networking devices.

F. *Operation of a Hybrid Network*

During the operation of a hybrid network, the system administrator shall:

• discover and identify new and rouge devices in the network environment,

• monitor network coverage,

• monitor interference, and identify potentially erroneous devices,

• monitor signal to noise ratio, and roaming capabilities,

• vulnerability scan and security test the network.

To automate these functions, the device manufacturers created WIPS (wireless intrusion prevention system) devices, which are easy to integrate into existing systems. To complete additional tasks, system administrators may use a portable measuring device that consist of both the necessary hardware and software components. An alternative measurement device can be a tablet running Android operating system, and equipped with the essential hardware components, with Analyzer and Surveyor applications. It is possible to measure with this device

• the signal strength, channel distribution, and channel interference of the available WLAN devices,

• weak signal level, and the incorrect coverage,

•in Survey model the tool is able to fit the measurement data with the corresponding layout.

III. Conclusion

With the proliferation of technology increases our wish to use technology. One possible implementation method is, if we create such working places, recreation areas and habitat that we can use our mobile devices with ease. Radio networks provide of this kind of solution, but we also shall see that this is a complex technology. The problems in a wireless network the administrators may face with various problems. System operators are expected to be able to collect as much information as possible about both the environment and the operated network. This approach assists to operate a complex WLAN network. After collecting, analyzing and filtering relevant information, it is possible to graphically display the result. Visualization is the tool to discover problems and errors in our complex system, and provide a way to improve the stability and robustness of the network operation solution.

References

[1] G. Ayres and J. Jones, "Self-mapping radio maps for location fingerprinting", *Wireless Networks*, vol. 21, no. 5, pp. 1485-1497, 2015.

[2] K. Lazányi, "Stressed Out by the Information and Communication Technologies of the 21st Century", *SCIENCE JOURNAL OF BUSINESS AND MANAGEMENT*, vol. 4, pp. 10-14 , 2016.

[3] Zs. Haig, "Információ - társadalom - biztonság", *NKE Szolgáltató Kft*, Budapest, 2015.

[4] Z. Weng, P. Orlik and K. J. Kim, "Classification of Wireless Interference on 2.4GHz Spectrum", *Wireless Communications and Networking Conference (WCNC)*, 2014.

[5] H. A. Iyad, A. A. Noor and S. Shamsul, "A light WLAN radio map for floor detection in multi-floor environment localization", *Software Engineering Conference (MySEC)*, Malaysian, 2015.

[6] L. Xiting, L. Banghui, N. Jianwei, S. Lei and C. Yuanfang, "HMF: Heatmap and WiFi Fingerprint-Based Indoor Localization with Building Layout Consideration", *Parallel and Distributed Systems (ICPADS)*, 2016.

Prediction of Electricity Consumption using Biologically Inspired Algorithms

Peter Halaš
Faculty of Informatics and
Information Technologies
Slovak University of Technology
Bratislava, Slovak Republic
xhalasp@stuba.sk

Marek Lóderer
Faculty of Informatics and
Information Technologies
Slovak University of Technology
Bratislava, Slovak Republic
marek_loderer@stuba.sk

Viera Rozinajová
Faculty of Informatics and
Information Technologies
Slovak University of Technology
Bratislava, Slovak Republic
viera.rozinajova@stuba.sk

***Abstract*—Prediction of electricity consumption has become a very investigated field of research recently. By lowering prediction error, we can minimize costs of suppliers. The classical approach using one prediction model has been proved insufficient. The problem is that none of the currently existing prediction models is sufficiently robust to estimate the forecast of time series accurately. One of the solutions to this problem is Ensemble learning. This approach attempts to create the best possible prediction by employing several different forecasting methods. We focused particularly on the ways of combining these methods. We apply different swarm intelligence algorithms to find the optimal combinations. The aim of this study is to compare prediction accuracy of the ensemble learning model to base forecasting methods. We also examine the optimal size of forecasting methods in the ensemble.**

***Index Terms*—time series prediction, ensemble learning, electricity consumption forecasting, biologically inspired algorithms**

I. Introduction

The introduction of smart meters, which are part of a smart metering system, opened access to data on energy consumption. These data open the possibility to apply various predictive models to reduce prediction error. Prediction of the observed variable, in this case, power consumption can be calculated by a number of models.

Various forecasting techniques have been studied in the literature. These techniques include Holt-Winters exponential smoothing [1], Linear and the Multiple Linear Regression [2], ARIMA models [3], Support Vector Regression [4], Neural Network models [5], Random Forest [6], Gradient Boosting [7] and Extremely Randomized Trees [8]. Each method reacts differently to changes of the data characteristics and therefore we cannot definitely say which one is the best. The number of existing methods is large, and it is difficult to orient in them. Because of these reasons, predicting the observed variable, using only one method can be unsafe.

Ensemble learning model uses a different strategy. Instead of relying on one (even an accurate forecasting method), it uses multiple methods for predicting. It consists of two [9] or three phases [10]. The first phase generates a set of base prediction models using one or more forecasting methods. In this phase, it is very important to take care of the diversity of the models. In the second phase the trimming is applied, which tends to increase the predictive power of ensemble model [11]. In the final phase, the prediction is calculated as combination of individual prediction models' results. In our study, we focus mainly on the last phase. We use swarm intelligence algorithms to combine the outputs of used prediction models.

The prediction accuracy of the ensemble model strongly depends on the accuracy of base models and on their diversity [12]. The diversity of base models can be achieved by two main approaches: homogeneous or heterogeneous [13], [14]. In heterogeneous approach, the ensemble uses several different types of base models [15], [16]. On the other hand, in homogeneous approach, the ensemble model usually consists of models of the same type, but trained on different subsets of available data [6].

The key part of the ensemble learning is the integration, which is performed as a combination of predictions of base models and given weights. The weights usually present the importance or performance of combined models. The weights can be computed by different methods, such as linear regression, gradient descent or by various optimization algorithms, e.g. particle swarm optimization or artificial bee colony [17], [18]. Ensemble learning was successfully used to predict values of time series [19] and electricity demand [20], [21].

This paper is organized as follows. In Section 2 we list used forecasting methods. In Section 3 we present our proposed Ensemble Learning Model with the use of the swarm intelligence algorithms. In Section 4 we present the results of experimental evaluation and the conclusion is in Section 5.

II. Forecasting Methods

Essential part of every ensemble is the base set of prediction models that are trained and subsequently combined in such a way that minimize the overall prediction error. In our study, we use several regression and time series forecasting methods to create models which are able to capture seasonal dependencies:

- *Holt-Winters exponential smoothing (HW)* [22] forecasts the future values of a time series as a weighted average of past values. It uses smoothing parameters for level, trend and seasonality component to calculate the weights for the observations.

- *Multiple Linear Regression (MLR)* [2] tries to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data.
- *Random Forest (RF)* [6] constructs a set of decision trees during training phase and outputs result as a mean prediction of the individual trees.
- *Support Vector Regression (SVR)* [23] attempts to find a linear regression function that can best approximate the actual output vector with a given error tolerance in a high dimensional feature space where the input data are mapped via a nonlinear function.
- *Gradient Boosting (XGBOOST)* [24] is a tree-based homogenous ensemble method. It produces new models to form an ensemble of weak models. Each next model is created to minimize the loss function calculated from the residuals of the previous ensemble.
- *Extremely Randomized Trees (EXRT)* [8] is an ensemble method that builds strongly randomized trees whose structures can be independent of the output values of the learning sample. It is caused by randomizing attributes and the cut-point choice in a tree node.
- *Exponential Smoothing with Seasonal and Trend decomposition using Loess (STL+ES)* [25] in the first step decomposes the original time series into three components: trend, season and noise. Then each component of time series is predicted by exponential smoothing method. Final prediction is made by a simple combination of the three predictions.
- *Neural Network (NN)* [26] is trained to learn the relationships between the input variables (past demand, time aspects, different day types, etc.) and historical load patterns.

III. Ensemble Learning Model

We propose the Ensemble Learning Model (or shortly Ensemble Model), for time series prediction with simple pruning and swarm intelligence algorithms. In our ensemble model, we apply both heterogeneous and homogeneous approach to generate prediction models. The pruning is used to reduce the number of redundant models and to increase the accuracy of final prediction. The main goal of pruning is to find a set of diverse models to achieve the maximum prediction accuracy. The combination of prediction models is carried out as a linear combination (a weighted mean), where the weights are calculated by swarm intelligence algorithms. It allows the weighting scheme to adapt over time, since the relative performance of prediction models can change.

The Ensemble model consists of three processes:

- Ensemble generation,
- Ensemble pruning,
- Ensemble integration.

These processes are be described in detail in the following sections.

A. Ensemble Generation

The prediction models can be generated by three approaches. The first is homogeneous approach where the prediction models are produced by the same forecasting method with different control input parameters and/or different training data. The second is heterogeneous approach which generates multiple models using multiple forecasting methods. The third is a combination of the previous described approaches.

In our proposed ensemble model, the last-mentioned approach is applied. There are three types of models generated for each forecasting method (see Figure 1). The first model is trained on all days of the week. The second model is trained only on working days. The third model is trained only on non-working days (holidays and weekends). The size of the training set for each model is determined experimentally.

The selection of these three types of model generation is important for the robustness of the ensemble model. Different models may react differently to changes in the development of time series. Each model is then trained using sliding window approach.

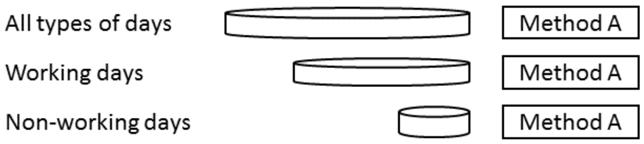


Fig. 1. Example of model generation by one prediction method.

B. Ensemble Pruning

The next step in the ensemble model is pruning, whose task is to select a subset of the generated models. There are several types of ensemble pruning methods that can be categorized as ranking-based, search-based and partitioning-based [11].

The proposed ensemble model uses ranking-based pruning where all prediction models are ordered according to a given evaluation function and then top N models are selected. This approach helps ensemble to produce more accurate forecasts by reducing high erroneous models.

C. Ensemble Integration

The last step of ensemble is called integration. The main goal is to combine outputs of prediction models into one final (possibly best) prediction. Ensemble integration in regression is usually defined as linear combination of given or calculated weights and predictions.

The ensemble integration process can be defined mathematically. Given the data of time series defined as:

$$Y = y_1, y_2, \dots, y_N \tag{1}$$

And given the prediction of i^th model, define as:

$$\hat{Y}^i = \hat{y}_1^i, \hat{y}_2^i, \dots, \hat{y}_N^i \tag{2}$$

Then the linear combination of the m predictions is obtained as:

$$\hat{Y}_{final} = \sum_{i=1}^{m} w_i \hat{Y}^i \tag{3}$$

The w_i is a weight assigned to the time series prediction $\hat{Y}^i$ of i^th prediction model. All weights must be nonnegative and the sum of all weights must be equal to 1.

For a combination of predictions, several methods are used. One method is an assignment of equal weights to each of the models. It is therefore a calculation of the average forecast from forecasts of all models. Another method is an assignment of different weights based on the prediction error of the models [27]. Since this is a problem of optimizing the weights of forecasts in order to reduce the prediction error, numeric optimization algorithms with constrains were applied. Namely: Ant Colony Optimization, Artificial Bee Colony and Firefly Algorithm, which belong to category of Swarm Intelligence Algorithms.

1) Ant Colony Optimization: Ant colony optimization (ACO) is inspired by the foraging behavior of ants in the collective search for food [28]. When an ant finds a food source, it evaluates it and carries some food back to the nest. On the way back to the nest, the ant releases a certain amount of pheromone on the ground. The amount of pheromone depends on the quantity and quality of the discovered food source. This information guides other ants to the food source.

Over time, if the pheromone trail is not reinforced by passing ants then the released pheromone on the path starts to evaporate, which reduces the path attractiveness for ants. Therefore, the pheromone communication not only guides other ants to the food source but enables them to find shortest (more frequently used) paths between the food source and their nest.

The main components of ACO algorithm are artificial ants and a pheromone model, which is used to probabilistically sample the search space. The artificial ants produce new solutions to the problem by iteratively adding part of the solution to their current solutions while considering the pheromone trail [28].

The original ACO algorithm can be applied only to solve discrete optimization problems. However, many real-world problems require continuous optimization. To address this issue, an ant colony optimization algorithm for continuous optimization (ACO$_{\mathbb{R}}$) was presented [29]. The algorithm stores a set of k best solutions, called solution archive, which represents the algorithm's *pheromone model*. Solutions are generated on a coordinate-per-coordinate basis using Gaussian kernel, which is defined as a weighted sum of several Gaussian functions [30].

2) Artificial Bee Colony: Artificial Bee Colony (ABC) is a Swarm Intelligence optimization algorithm which simulates the foraging behavior of honeybees [18]. The algorithm includes three types of bees in the colony, namely foragers, onlookers and scouts. Each bee type performs different activity for the colony. Each employed forager is associated with one specific food source (solution). The source has a form of a d-dimensional vector $(X_i = (x_1, x_2, x_3, \ldots, x_d)$.

The task of an employed forager is to search for new food source V_i with more nectar in the neighborhood of the current food source. The search is defined by:

$$V_{ij} = X_{ij} + \varphi_{ij} \times (X_{ij} - X_{kj}) \tag{4}$$

where $k \in \{1, 2, \ldots, BN\}$ is selected randomly and $(k \neq i)$, BN is the number of employed bees, X_j is randomly selected candidate solution $(i \neq j)$, $j \in \{1, 2, \ldots, N\}$ is a randomly chosen index, and φ_{ij} is a random number from a range [-1,1].

If the fitness of the new searched source V_i is better than the fitness of current source X_i then the employed forager immediately forgets the current source X_i and memorizes the new one.

Each forager carries information about quality and intensity of the food source and its distance from the hive. The information about all sources are transmitted to onlookers in the dancing area. Each onlooker chooses one food source with a probability, which is calculated as follows:

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \tag{5}$$

where SN is the number of food sources and $fitness_i$ is the fitness value of the i^{th} solution.

Thus, the solution with a higher value of the fitness function has a higher chance to be selected.

In case a food source is not improved over a predefined number of trials (called *limit*), then the food source is exhausted and abandoned. The employed bee whose food source has been exhausted turns into a scout. The scout bee searches for new source regardless of existing sources. The search for new source X_i can be formulated as:

$$X_i^j = X_{min}^j + rand(0,1)(X_{max}^j - X_{min}^j) \tag{6}$$

where $j \in \{1, 2, \ldots, N\}$ are randomly chosen indexes.

As a result of such a behavior, the scouts have a low search cost and can occasionally discover rich unknown food sources.

3) Firefly Algorithm: The Firefly algorithm is a swarm intelligence optimization algorithm inspired by the luminescent flashlight behavior of fireflies [31]. The algorithm uses a population of agents (known as fireflies) which communicate with each other via bioluminescent flashlight and iteratively explore cost function space of a considered optimization problem.

The Firefly Algorithm works by three rules [31]:

1) Each firefly is unisexual, so that if one attracts the other, it does not depend on its sex.
2) The degree of attractiveness is proportional to the intensity of firefly's brightness. If the distance between two fireflies increases, then the intensity of brightness decreases due to the light absorption of air. The less bright fireflies always move toward brighter one. If a firefly cannot see any brighter one than itself, it moves randomly.

3) The brightness of firefly is determined by the value of the objective function of a given problem.

An effective and accurate firefly algorithm requires a proper formulation of light intensities I and attractiveness β. For simplicity, it can be assumed that the firefly attractiveness is determined by its brightness which is associated with the encoded objective function (see rule 3). As mention in the rule 2, the attractiveness (through brightness) also changes by distance between firefly i and firefly j. In the simplest formulation, the final light intensity $I(r)$ can be a function of source light intensity I_s and a distance r between two objects (the inverse square law) (7).

$$I(r) = I_s/r^2 \tag{7}$$

Considering an environment with certain light absorption ability (for example air), the light intensity I varies with the distance r and a fixed light absorption coefficient γ. It can be formulated as:

$$I = I_0 e^{-\gamma r^2} \tag{8}$$

where I_0 is the original light intensity.

If firefly's attractiveness is considered proportional to the light intensity seen by adjacent fireflies, the I variable in (8) can be substituted by attractiveness , so the equation for firefly attractiveness is defined as:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{9}$$

where $_0$ is the attractiveness at $r = 0$. The distance between two fireflies i and j (at x_i and x_j) is calculated as the Cartesian distance (10).

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{10}$$

where $x_{i,k}$ is the k^{th} component of the spatial coordinate x_i of i^{th} firefly and d is number of components. The movement of less attractive firefly to more attractive firefly is defined as:

$$x_i = x_i + \beta_0 e^{-r^2} (x_j - x_i) + \alpha \epsilon_i \tag{11}$$

where the second term is used to express attractiveness and the third term is a random factor α with a vector of random numbers from Gaussian distribution ϵ_i.

IV. Experimental Evaluation

A. Performance Metric

To evaluate the experiments, we used the metric MAPE. Its advantage is that it is independent of the unit in which the data is measured. The downside is that it cannot be used if the time series contains zero values. It is computed as follows:

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{\hat{Y}_i - Y_i}{Y_i} \right| \tag{12}$$

where N is the number of the samples. Y_i is the actual value and $\hat{Y}_i$ is the forecast value.

B. Data

The proposed ensemble model was evaluated on three energy load consumption datasets. The first dataset consists of 30 minutes time series measurements from Australian Energy Market Operator (AEMO)[1]. Data are aggregated for each state of Australia. In the experiment, we used data from the New South Wales, since it is the most populated Australian state with high electricity demand.

The second dataset consists of 5 minutes time series measurements from The New York Independent System Operator (NYISO)[2]. We used aggregated data from the area of New York City.

The third dataset consists of hourly power load consumption measurements from Ontario, Canada. The power system is managed by the Independent Electricity System Operator (IESO)[3].

C. Experimental Settings

In our experiments, we focused on the accuracy of proposed ensemble model using different swarm intelligence algorithms. The experiments were also designed to determine the optimal number of prediction models in the ensemble.

Prediction models were trained throughout the year 2008 on all datasets. The aim of this training phase was to determine the optimal size of the training set for each model, with respect to minimization of MAPE metric.

The Table I shows the number of training days for each prediction model for AEMO dataset. The first column lists the number of days used to train the models on all types of days, in the second column models were trained only on working days and the last column lists the models trained on non-working days.

TABLE I
Size of the optimal number of training days for forecasting models on AEMO dataset.

Model name	All types of day	Working days	Non-working days
ARIMA	21	10	6
EXTR	7	3	5
HW	8	15	8
MLR	14	15	7
NN	7	4	2
RF	8	8	4
STL+ES	8	6	3
SVR	21	4	2
XGBOOST	11	6	8

After the optimal number of training days for each prediction model was estimated, we used the sliding window approach to perform daily predictions for whole year 2009 on all tested datasets. Subsequently, we calculated weights for the combination of models' predictions. We used three different swarm intelligence algorithms to compute the weights

[1] http://www.aemo.com.au

[2] http://www.nyiso.com

[3] http://www.ieso.ca/

for each prediction model. Simultaneously, we were testing optimal number of prediction models for the ensemble model.

In our experiments, we set the number of iteration for ABC to 50, the population composed of 20 individuals and the food limit was 200. In FA the number of iteration was 100, number of individual was set 50, the light absorption coefficient was set to 0.6, base attraction was 0.1, base attraction was set to 0.5 and random move factor was 0.1. In $ACO_{\mathbb{R}}$, the ant colony consisted of 30 individuals over 30 iterations, the intensification factor q was set to 0.9 and Deviation-Distance Ratio zeta was 1.

The dimension of all individuals was given by the number of base models N. All swarm intelligence algorithms used MAPE as a loss function, which they try to minimize. The value of MAPE was calculated on an 8-day window of previous data. The size of the window was determined experimentally.

D. Results

Table II contains average daily prediction error of used prediction models. The table also shows the lowest prediction error of three tested ensemble models with different integration methods. The results show that all ensemble models achieved better prediction results than base models in every test case. These results also show that the majority of prediction models benefit from splitting and predicting separately working and non-working days. But the increase of the prediction accuracy is not significantly high.

The prefix ”-split” means that the model was trained on separate working and non-working days. Values in bold represent the minimal prediction error for each test case of both ensemble models and base prediction models.

TABLE II
AVERAGE DAILY MAPE PREDICTION ERROR OF FORECASTING MODELS AND ENSEMBLE MODELS ON TESTED DATASETS.

Model name	AEMO	NYISO	IESO
ARIMA	11.686	20.404	10.230
ARIMA - split	11.363	22.825	10.998
HW	11.227	16.773	4.655
HW - split	8.500	7.476	4.754
EXTR	4.846	5.463	5.020
EXTR - split	**3.957**	5.034	4.130
MLR	4.889	5.703	5.029
MLR - split	4.567	5.330	5.483
NN	5.222	6.260	16.686
NN - split	5.000	6.412	9.547
RF	4.802	5.754	4.931
RF - split	4.314	5.279	5.184
STL+ES	5.652	4.790	6.141
STL+ES - split	4.157	**4.429**	**3.443**
SVR	5.096	5.503	7.242
SVR - split	4.533	5.248	4.798
XGBOOST	4.889	5.466	5.105
XGBOOST - split	3.992	5.095	4.002
ABC	3.704	4.096	3.293
$ACO_{\mathbb{R}}$	3.726	4.085	3.290
FA	**3.677**	**4.061**	**3.248**

The best prediction accuracy among the base modes was achieved by Exponential Smoothing with Seasonal and Trend decomposition using Loess. This model was trained on separated working and non-working days. The best prediction accuracy among ensemble models was obtained by the ensemble model with firefly algorithm. But the differences among this model and the other two ensemble models is not very significant.

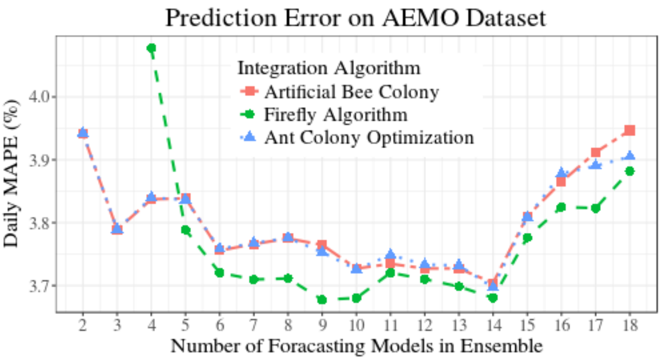


Fig. 2. Progress of the average daily MAPE prediction error of ensemble models on AEMO dataset.

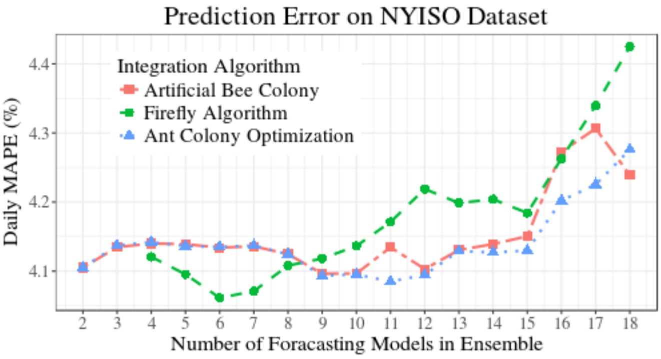


Fig. 3. Progress of the average daily MAPE prediction error of ensemble models on NYISO dataset.

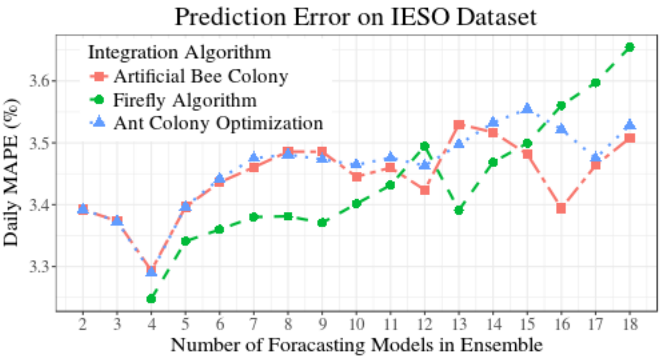


Fig. 4. Progress of the average daily MAPE prediction error of ensemble models on IESO dataset.

Figures 2, 3 and 4 show the progress of the average daily prediction error of tested ensemble model with different swarm intelligence algorithms. According to the results, all ensemble models favored from lesser number of prediction models. It is probably duet to fact, that all prediction models had been ordered by their performance in previous year and therefore each next model added into ensemble contained less accurate results. The results also showed that ensemble model with FA and very small set of forecasting models (2 or 3) always achieved high prediction error. In some cases, the error exceeded 15% MAPE.

We excluded the first two results of the firefly algorithm from displaying to make the figures more readable.

V. Conclusions

We have analyzed, designed and verified own ensemble model for time series prediction with simple pruning and swarm intelligence algorithms. We selected several prediction models in the phase of generating and these models have been trained on specific size of training data. We also used well-known methods like similar day approach (distinguishing working and non-working days) and sliding window to obtain more accurate daily prediction results. We tested our solution on three publicly available electricity consumption datasets.

In the process of integration, we compared weighted schemes computed by three swarm intelligence algorithms. Our goal was to minimize the error of the proposed ensemble model. The results showed that ensemble model with all three swarm intelligence algorithms outperformed all base prediction models. The best results were achieved by Firefly Algorithm with small set of base prediction models in ensemble, but the minimal size of the set should be at least 4. The other two tested swarm intelligence algorithms achieved slightly worse results as the firefly algorithm.

The aim of future work will be to improve the pruning phase by dynamic selection of base models based on their performance in a given recent time period.

Acknowledgment

This work was partially supported by the Slovak Research and Development Agency under the contract APVV-16-0213 and ITMS 26240120039, co-funded by the ERDF.

References

[1] Y. H. Guo, X. P. Shi, and X. D. Zhang, "A study of short term forecasting of the railway freight volume in china using arima and holt-winters models," in *2010 8th International Conference on Supply Chain Management and Information*, Oct 2010, pp. 1–6.

[2] V. Kantikoon and V. Kinnares, "The estimation of electrical energy consumption in abnormal automatic meter reading system using multiple linear regression," in *2013 IEEE International Conference on Electrical Machines and Systems (ICEMS)*, Oct 2013, pp. 826–830.

[3] W.-C. Hong, *Intelligent Energy Demand Forecasting (Lecture Notes in Energy)*. Springer, 2015.

[4] V. Kecman, *Support Vector Machines – An Introduction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–47.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[8] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Mar 2006.

[9] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, *Dynamic Integration of Regression Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 164–173.

[10] F. Roli, G. Giacinto, and G. Vernazza, *Methods for Designing Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 78–87.

[11] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–40, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2379776.2379786

[12] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, May 2003.

[13] S. Bian and W. Wang, "Investigation on diversity in homogeneous and heterogeneous ensembles," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, July 2006, pp. 3078–3085.

[14] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," *Journal Of Big Data*, vol. 1, no. 1, p. 5, Jun 2014.

[15] A. C. Palaninathan, X. Qiu, and P. N. Suganthan, "Heterogeneous ensemble for power load demand forecasting," in *2016 IEEE Region 10 Conference (TENCON)*, Nov 2016, pp. 2040–2045.

[16] G. Dudek, "Heterogeneous ensembles for short-term electricity demand forecasting," in *2016 17th International Scientific Conference on Electric Power Engineering (EPE)*, May 2016, pp. 1–6.

[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[18] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108 – 132, 2009.

[19] J. D. Wichard and M. Ogorzalek, "Time series prediction with ensemble models," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, July 2004, pp. 1625–1630 vol.2.

[20] W. Shen, V. Babushkin, Z. Aung, and W. L. Woon, "An ensemble model for day-ahead electricity demand time series forecasting," in *Proceedings of the Fourth International Conference on Future Energy Systems*, ser. e-Energy '13. New York, NY, USA: ACM, 2013, pp. 51–62.

[21] L. Xiao, J. Wang, R. Hou, and J. Wu, "A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting," *Energy*, vol. 82, no. Supplement C, pp. 524 – 549, 2015.

[22] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on european data," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2213–2219, Nov 2007.

[23] F. Zhang, H. Dai, and D. Tang, "A conjunction method of wavelet transform-particle swarm optimization-support vector machine for streamflow forecasting," *Journal of Applied Mathematics*, vol. 2014, pp. 1–10, 2014.

[24] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 10 2001.

[25] R. B. Cleveland, W. S. Cleveland, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, p. 3, 1990.

[26] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*. IEEE, dec 2014.

[27] R. Adhikari and R. K. Agrawal, "Performance evaluation of weights selection schemes for linear combination of multiple forecasts," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 529–548, Dec 2014.

[28] M. Dorigo and T. Sttzle, *Ant Colony Optimization (MIT Press)*. A Bradford Book, 2004.

[29] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1155 – 1173, 2008.

[30] K. Socha and C. Blum, "An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training," *Neural Computing and Applications*, vol. 16, no. 3, pp. 235–247, May 2007.

[31] X.-S. Yang, *Firefly Algorithms for Multimodal Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 169–178.

Superimposed Multidimensional Schemas for RDF Data Analysis

Median Hilal, Christoph G. Schuetz, Michael Schrefl
Johannes Kepler University Linz, Austria
{hilal, schuetz, schrefl}@dke.uni-linz.ac.at

***Abstract*—Traditional data analysis employs online analytical processing (OLAP) systems operating on multidimensional (MD) data. The Resource Description Framework (RDF) serves as the foundation for the publication of a growing amount of semantic web data still largely untapped by data analysis. RDF data, however, do not typically follow an MD structure and, therefore, elude traditional OLAP. We propose an approach for superimposing MD structures over arbitrary RDF datasets. On top of that, we present a high-level querying mechanism to express MD queries, which can be automatically translated into SPARQL queries over the source data. As a consequence, data analysts that are unfamiliar with SPARQL may still incorporate RDF data sources into the analysis. Superimposed MD schemas also serve as foundation for Semantic Web Analysis Graphs which capture analysis processes for increased self-service capabilities.**

I. Introduction

Online analytical processing (OLAP) systems represent data in multidimensional (MD) format that allows analysts to view the data at different granularities in order to gain insights (see [1] for a comprehensive introduction). MD schemas typically consist of facts, measures, and dimensions with multiple levels. Facts are the subjects of interest whereas measures quantitatively describe the facts. Dimensions are organized hierarchically into levels, which allows for aggregating measures on different granularities.

Although OLAP systems traditionally operate on enterprise internal data, external (web) data are increasingly becoming an important source for data analysis. An increasing amount of external data are published in Resource Description Framework (RDF) format as sets of Linked Open Data (LOD, see [2]). These RDF data present a prime source of data for the analysis to provide analysts with useful insights.

The exploitation of external RDF data presents several issues. First, RDF data do not typically have a structure suitable for data analysis but rather follow a heterogeneous and semi-structured data model that is not under the analyst's control. Traditional data integration of RDF data into a data warehouse is often impractical due to velocity and volume as well as the decentralized nature of the RDF data sources. Consequently, a method that enables expressing analytical structures [3], [4], ideally multidimensional, on top of these RDF data is of high importance. Second, even if the data are organized in MD format, conducting analysis over the RDF data is still not a straight-forward task for casual analysts. Without an accompanying querying mechanism in place, exploiting the MD structure would still require knowledge of an RDF query language such as SPARQL. Third, in today's dynamic and open nature of business, casual analysts are in more need of making time-critical decisions [5] with less specialists' support. Therefore, providing casual analysts with self-service tools to facilitate their tasks becomes crucial.

In this paper, we introduce an approach for expressing superimposed MD schemas over arbitrary RDF data sources. We extend analytical schemas [3] with an explicit and expressive MD structure that captures the hierarchically-organized dimensions; the MD structures can be published and reused. On top of superimposed MD schemas, we adapt the notion of analysis situation [6], a mechanism of expressing dynamic templates to model families of MD queries corresponding to specific business interests. Consequently, analysts are assisted with analyzing the RDF data while being spared the technical details. Only MD elements are involved in the query, and SPARQL queries corresponding to an instantiation of an analysis situation can be automatically generated. Furthermore, we employ the superimposed multidimensional schemas and the adapted analysis situations as a foundation for an elaborated self-service vision of OLAP analysis over RDF data sources. Specifically, we sketch the semantic web analysis graphs which pro-actively model the knowledge about the analytical process in response to a business need or goal.

As a motivational example, consider *Wikidata*[1], a free and open knowledge base, which provides a SPARQL query service[2]. Wikidata includes data about films, where a *film* has a *duration*, *director*, *publication date*, *country of origin*, *genre*, *main subject* etc. The Wikidata schema, however, may seem quite complex for analysts not familiar with the schema. Now consider *Mia*, a data journalist working for a media outlet. Mia is preparing an analytical report about movies in 2017. Specifically, Mia needs to identify and compare the number of films and their average durations classified by directors, genres, and countries of origin in the second half of 2017. Furthermore, she wants to manipulate the results depending on specific criteria. Mia, however, cannot perform her task without sufficient knowledge of the SPARQL query language and the semantics and structure of film data in Wikidata. Even if Mia has these skills, the task is complicated, misleading,

Median Hilal is funded by Erasmus Mundus - ASSUR program.

[1] The running example in this paper works on Wikidata schema and instance data available at https://www.wikidata.org/ as of 17-Sep-2017.

[2] https://query.wikidata.org/

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

and likely to be unfeasible. Nonetheless, the superimposition of a suitable MD schema over the film data renders these data accessible to OLAP-style analysis needed by Mia. Furthermore, the definition of analysis situations that Mia can instantiate to suit her needs will render Mia's task feasible.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the multidimensional modeling using the superimposed schemas. Section IV presents the multidimensional analysis using the superimposed schemas. Section V discusses the presented approach and gives an outlook on future work.

II. Related Work

We classify related literature on RDF analytics into three main categories.

Ontology-based local data warehousing. Nebot and Berlanga [7] propose an approach to enable warehousing semantic web data that correspond to ontologies. Semantic data that meet analysis requirements are extracted through a complex process and locally stored. Afterwards, classical OLAP can be performed on the data. The frequent changes, volume, and complexity of semantic web data impose limitations on the approach.

RDF vocabularies. Some methods utilize RDF(S) vocabularies to allow representing the schema and instances of OLAP cubes and implementing OLAP operations. Most-prominent vocabulary is arguably the data cube (QB) [8] vocabulary proposed by the W3C to publish multidimensional data, especially statistics, on the web in such a way that the data can be linked to related data sets. Kämpgen et al. [9] investigate performing OLAP queries via SPARQL on an RDF store to interact with statistical linked data. QB vocabulary is not originally designed for OLAP [10]. As a consequence, the QB4OLAP vocabulary [10] extends QB in order to support OLAP while remaining compatible with QB. The Cube Query Language (CQL) [11] enables high-level OLAP querying of QB4OLAP data cubes. These methods are only applicable to data already represented and published using a corresponding vocabulary. As a consequence, they fall short in addressing a wide variety of analytical possibilities in non-statistical RDF data sources.

Analytical schemas. Colazzo et al. [3] propose a method for enabling analysis over RDF data by superimposing analytical schemas. Analytical schemas are graphs where nodes and edges are views over the data and have corresponding analytical instances, and can be queried by means of rooted basic graph patterns. Colazzo et al. [3], however, do not elaborate on multidimensional structure, dimension hierarchies, or employing such structures for the purpose of multidimensional querying.

III. Multidimensional Modeling

In this section, we introduce a multidimensional (MD) modeling approach for RDF data sources using Superimposed Multidimensional Schemas (SMDS). Superimposed multidimensional schemas are laid over RDF data sources, allowing for the expression of multidimensional perspectives, basically following the dimensional-fact model [12]. An SMDS is basically a set of views mapped to the RDF data sources through SPARQL queries. Consequently, the RDF data themselves are not affected.

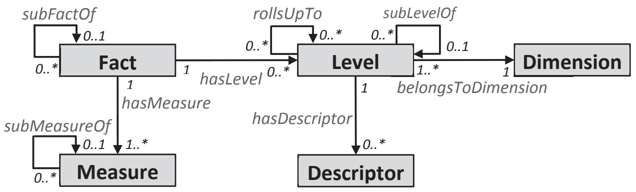


Fig. 1. SMDS metamodel

A. Metamodel

Figure 1 depicts the SMDS metamodel[3] which can be expressed as an OWL ontology. The metamodel can be instantiated to express MD schemas for particular RDF sources. *Fact, Level, Measure, Descriptor,* and *Dimension* are the MD elements of the metamodel. Instances of these elements can be connected via the MD properties *hasLevel, hasMeasure, hasDescriptor, rollsUpTo, belongsToDimension, subFactOf, subLevelOf,* and *subMeasureOf*. An instance of *Fact* represents the subject of interest which is quantified by measures, represented by the instances of *Measure*. The instances of *Level* are used to model the level of granularity that the measures for a subject of interest are captured at. The levels belong to a *Dimension* and can be further described by descriptors. *hasLevel* and *hasMeasure* connect facts to their levels and measures respectively. *hasDescriptor* connects a level to its descriptors while *belongsToDimension* connects a level to the dimension it belongs to. *rollsUpTo* expresses roll-up hierarchal relationships between levels. *subFactOf*, *subMeasureOf*, and *subLevelOf* introduce a specialization mechanism to reflect subsumption relationships in the original RDF data and cope with the heterogeneity of RDF data by allowing specialized elements to have their own properties. Furthermore, these specializations propagate the multidimensional knowledge from the super elements to their sub elements.

Figure 2 shows an example instantiation of the SMDS metamodel for the films scenario. *Film* is the fact and represents the type of entities which are used to calculate the measures' values. *Direction, GenreDim, Origin,* and *PublicationTime* are the dimensions, where *Direction, Origin,* and *PublicationTime* have levels of granularity. On *Direction* dimension, *Director* rolls up to each of *Gender* and *Citizenship*. *Director* is further described by the descriptor *BirthDate*. On *Origin* dimension, *Country* rolls up to *Continent*. Finally, on *PublicationTime* dimension, the publication date *Date* rolls up to the year of publication *Year*. *NumOfFilms* and *Duration* are the measures. *NumOfFilms* is the number of films, a special kind of measure that corresponds to the measure in *empty facts* [12]. The measures can be rolled up along the hierarchies of the dimensions, e.g., *Duration* is available by *Date* and *Director*, and then can be available

[3] We do not mention the commonsense constraints on the MD schema.

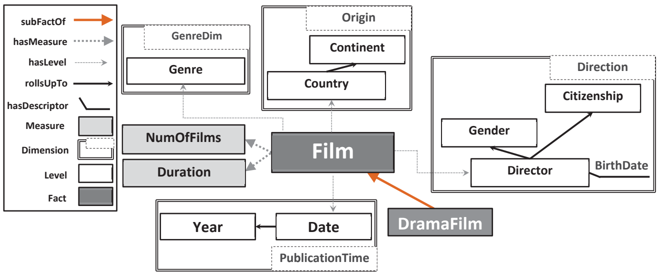


Fig. 2. The SMDS instantiation for *Wikidata* films scenario

by *Year* and *Citizenship* of the director. *Film* and *DramaFilm* illustrate a specialization, where *DramaFilm* is *subFactOf Film*, thus inheriting its MD knowledge, e.g., it has *Duration* as a measure. Furthermore, *DramaFilm* may have its own MD properties, e.g., it may have a *DramaType* additional level.

We formalize an SMDS as a graph; we adapt and extend definitions introduced by Colazzo et al. [3]. In particular, we extend the definition of the analytical schema to become the multidimensional graph MDG in order to handle multidimensional hierarchical structure[4].

Definition 1: An SMDS corresponds to a graph $MDG = < N_{MD}, E_{MD}, U_{MDG}, Q, \pi, \lambda, \delta >$ (adapted and extended from the notion of analytical schemas [3]) where:

- N_{MD} is the set of nodes of the graph, representing the MD elements.
- $E_{MD} \subseteq N_{MD} \times N_{MD}$ is the set of edges of the graph, representing the direct MD relationships between the MD elements.
- U_{MDG} is a set of URIs.
- Q is a set of mapping queries.
- $\pi : N_{MD} \cup E_{MD} \rightarrow R \cup P$ is a function that maps each node to its MD role in $R = \{fact, level, measure, descriptor, dimension\}$, and each edge to its MD property in $P = \{hasLevel, hasMeasure, hasDescriptor, rollsUpTo, subFactOf, subLevelOf, subMeasureOf, belongsToDimension\}$. The usage of edges to connect nodes corresponds to the metamodel[5] shown in Fig. 1.
- $\lambda : N_{MD} \cup E_{MD} \rightarrow U_{MDG}$ is bijective naming function that assigns a unique name (URI) to each MD node and each edge.
- $\delta : (N_{MD} \setminus N_{MD_{dim}}) \cup E_{mapped} \rightarrow Q$ is a function that defines the mappings of nodes and edges of the graph to the source schema. $N_{MD_{dim}}$ is the set of the MD nodes the roles of which are dimensions, i.e., $N_{MD_{dim}} = \{n_d : n_d \in N_{MD} \; and \; \pi(n_d) = dimension\}$. $P_{mapped} = \{hasLevel, hasMeasure, hasDescriptor, rollsUpTo\} \subset P$ is the set of the MD properties that have mappings. Correspondingly, $E_{mapped} \subset E_{MD}$ is the set of edges where the MD property that the edge represents is in P_{mapped}, i.e., $E_{mapped} = \{e : e \in E_{MD} : \pi(e) \in P_{mapped}\}$. See Sect. III-B for the more detailed description of the mappings.

Definition 2: We use unary predicates applied to URIs from U_{MDG} and binary predicates applied to pairs of URIs from U_{MDG} to define MD concepts over MDG as follows:

- Consider λ^{-1} the inverse function of λ. Given $x \in U_{MDG} : \lambda^{-1}(x) \in N_{MD}$, we denote the node[6] corresponding to x as n_x, i.e., $n_x = \lambda^{-1}(x)$.
- The predicate $Fact(f)$ is *true if and only if* $\pi(n_f) = fact$. Similarly, $Level(l)$ is *true if and only if* $\pi(n_l) = level$. $Measure(m)$, $Descriptor(dr)$, and $Dimension(d)$ are defined analogously.
- A fact $Fact(f)$ has a measure $Measure(m)$, denoted as $HasMeasure(f, m)$, *if and only if* $\exists e = (n_f, n_m) \in E_{MD} : \pi(e) = hasMeasure$. $HasDescriptor(l, dr)$, $BelongsToDimension(l, d)$ are defined analogously.
- A roll-up relationship may be direct or indirect. A level $Level(l_1)$ rolls up *directly* to another level $Level(l_2)$, denoted as $RollsUpDirectlyTo(l_1, l_2)$, *if and only if* $\exists e = (n_{l_1}, n_{l_2}) \in E_{MD} : \pi(e) = rollsUpTo$. The general case is that a roll-up relationship might be direct or indirect, which is denoted by the predicate $RollsUpTo$. A level $Level(l_1)$ rolls up to another level $Level(l_m)$, denoted as $RollsUpTo(l_1, l_m)$, *if and only if* there is a sequence of edges in MDG $((n_{l_1}, n_{l_2}), (n_{l_2}, n_{l_3}), ..., (n_{l_{m-1}}, n_{l_m}))$, where $m \geq 2$, such that: $\forall (n_{l_i}, n_{l_{i+1}})$, for all $1 \leq i < m$, $\pi((n_{l_i}, n_{l_{i+1}})) = rollsUpTo$.
- Each of $SubFactOf$, $SubLevelOf$, and $SubMeasureOf$ are defined analogously to $RollsUpTo$, with the possibility to be direct or indirect.
- A fact may, similarly, have direct or indirect levels; the direct case is when the level is a base level[7]. A fact $Fact(f)$ has a *base* level $Level(l)$, denoted as $HasBaseLevel(f, l)$, *if and only if* $\exists e = (n_f, n_l) \in E_{MD} : \pi(e) = hasLevel$. A fact $Fact(f)$ has a level $Level(l_m)$, denoted as $HasLevel(f, l_m)$, *if and only if* $(HasBaseLevel(f, l_m))$ or $(\exists l_1 : Level(l_1)$ and $HasBaseLevel(f, l_1)$ and $RollsUpTo(l_1, l_m))$.[8]

Example 1: In Fig. 2, where the URIs of the MD nodes are shown, n_{Film} and $n_{Duration} \in N_{MD}$. Similarly, edges $(n_{Film}, n_{Director})$ and $(n_{Film}, n_{Date}) \in E_{MD}$. Also $\pi(n_{Film}) = fact$, $\pi(n_{Director}) = level$, $\pi((n_{Film}, n_{Duration})) = hasMeasure$, and $\pi(n_{Origin}) = dimension$. Furthermore, the following predicates hold: $Fact(Film)$, $Measure(Duration)$, $HasMeasure$ $(Film, Duration)$, $HasBaseLevel$ $(Film, Director)$, and $HasLevel$ $(Film, Gender)$.

[4] In the following, we informally say *MD elements* to denote instances of the MD elements of the metamodel.

[5] For example, given an edge $e = (n_1, n_2) \in E_{MD}$ and $\pi(e) = hasLevel$, then $\pi(n_1) = fact$ and $\pi(n_2) = level$.

[6] By convention, we use the letters *f* for facts, *l* for levels, *m* for measures, *d* for dimensions, and *dr* for descriptors. We use the same convention for the corresponding nodes, e.g., n_f for nodes the MD role of which is fact.

[7] Corresponding to the finest granularity level on a specific dimension.

[8] Note that the direct cases are covered by the indirect ones.

TABLE I
RULES FOR PROPAGATING THE MD KNOWLEDGE FROM SUPER MD ELEMENTS TO THEIR SUB ELEMENTS.

$SubFactOf(f', f) \wedge HasBaseLevel(f, l) \rightarrow HasBaseLevel(f', l)$
$SubFactOf(f', f) \wedge HasMeasure(f, m) \rightarrow HasMeasure(f', m)$
$SubLevelOf(l', l) \wedge RollsUpDirectlyTo(l, l_1) \rightarrow RollsUpDirectlyTo(l', l_1)$
$SubLevelOf(l', l) \wedge HasDescriptor(l, dr) \rightarrow HasDescriptor(l', dr)$

The direct MD relationships resulting from propagating the MD knowledge from super MD elements to their sub elements should exist in MDG. The rules shown in Table I can be applied to propagate this MD knowledge, i.e., infer new edges in MDG. The mappings and the names of the inferred edges, however, should be added.

B. Mappings

In the following we adapt and extend definitions introduced by Colazzo et al. [3] for mapping analytical schemas to source data. A *Triple Pattern (TP)* is an RDF triple of subject, predicate, and object where each can be a variable. A *Basic Graph Pattern (BGP)* is a set of triple patterns [13]. We use the *Conjunctive Query (CQ) notation* [3] of the form $q(\bar{x})$:- $t_1, ..., t_\alpha$ where $\{t_1, ..., t_\alpha\}$ is a BGP. Head variables $\bar{x}$ are called distinguished variables and denoted as $HeadVars(q)$ and are a subset of the variables at the body. We denote the body of the query as $Body(q)$ which is $t_1, ..., t_\alpha$. The *Join Query (JQ)* [3] of a set of BGP queries $q_1, ..., q_n$, with their non-distinguished variables being pairwise disjoint, is defined by a new BGP query $q(\bar{x})$:- $q_1(\bar{x_1}) \wedge ... \wedge q_n(\bar{x_n})$ where $\bar{x} \subseteq \bar{x_1} \cup ... \cup \bar{x_n}$ and its answer set is defined by the answer set of the BGP $q^{\bowtie}(\bar{x})$ resulting from *combining* the bodies $q^{\bowtie}(\bar{x})$:- $Body(q_1(\bar{x_1}))$,..., $Body(q_n(\bar{x_n}))$. In the following, we adapt the the mapping function and the mapping convention introduced by Colazzo et al. [3] to fit our context.

Definition 3: The *mapping function* $\delta : (N_{MD} \setminus N_{MD_{dim}}) \cup E_{mapped} \rightarrow Q$ assigns to nodes unary BGP queries, and to edges binary BGP queries.

Definition 4: The *mapping convention* is that the mapping query of a node specifies the instances that belong to the node, while the mapping query of an edge $e = (n_1, n_2)$ has two head variables one for each node, specifying how the instances of n_1 and n_2 are connected via the edge which means that $HeadVars(\delta(e)) = HeadVars(\delta(n_1)) \cup HeadVars(\delta(n_2))$. The names of the non-distinguished variables among the mapping queries are pairwise disjoint. The names of the distinguished variables expressing the same concept are identical.

Example 2: Table II shows the mappings of some of the MD nodes and edges from Fig. 2 following the mapping convention. Variables' names start with a question mark (*?*). Note that the prefixes *wdt*[9] and *wd*[10] used in the example are specific to *Wikidata*[11]. The unary queries in Table II are determining the instances belonging to the corresponding class using *wdt:P31* which stands for *Wikidata instanceOf* relationship[12]. For example, $q_1(?film)$ queries the films ($?film$) that are instances ($wdt{:}P31$) of *Film* in *Wikidata* ($wd{:}Q11424$). Regarding the binary queries, $q_8(?film, ?director)$, for example, queries the values of $?film$ and $?director$, that are related via *director* relationship ($wdt{:}P57$).

TABLE II
AN EXCERPT OF THE MAPPINGS OF THE MD NODES AND EDGES

Node/Edge	Mapping
n_{Film}	$q_1(?film)$:- $?film\ \ wdt{:}P31\ \ wd{:}Q11424$
$n_{Country}$	$q_2(?country)$:- $?country\ \ wdt{:}P31\ \ wd{:}Q6256$
$n_{Director}$	$q_3(?director)$:- $?director\ \ wdt{:}P31\ \ wd{:}Q5$
n_{Gender}	$q_4(?gender)$:- $?gender\ \ wdt{:}P31\ \ wd{:}Q48277$
$n_{Citizenship}$	$q_5(?citizenship)$ -: $?citizenship\ \ wdt{:}P31\ \ wd{:}Q6256$
n_{Genre}	$q_6(?genre)$:- $?genre\ \ wdt{:}P31\ \ wd{:}Q201658$
$(n_{Film}, n_{Country})$	$q_7(?film, ?country)$:- $?film\ \ wdt{:}P495\ \ ?country$
$(n_{Film}, n_{Director})$	$q_8(?film, ?director)$:- $?film\ \ wdt{:}P57\ \ ?director$
$(n_{Director}, n_{Gender})$	$q_9(?director, ?gender)$:- $?director\ \ wdt{:}P21\ \ ?gender$
$(n_{Director}, n_{Citizenship})$	$q_{10}(?director, ?citizenship)$:- $?director\ \ wdt{:}P27\ \ ?citizenship$
$(n_{Director}, n_{BirthDate})$	$q_{11}(?director, ?birthDate)$:- $?director\ \ wdt{:}P569\ \ ?birthDate$
(n_{Film}, n_{Genre})	$q_{12}(?film, ?genre)$:- $?film\ \ wdt{:}P136\ \ ?genre$
$(n_{Film}, n_{Duration})$	$q_{13}(?film, ?duration)$:- $?film\ \ wdt{:}P2047\ \ ?duration$
(n_{Film}, n_{Date})	$q_{14}(?film, ?date)$:- $?film\ \ wdt{:}P577\ \ ?date$

Definition 5: The *MD Path (MDP)* is any directed path p in MDG of the form $(n_1, (n_1, n_2), n_2, ..., n_m)$, where $m \geq 2$ and all the nodes and edges have mappings. The path p is fact-rooted if and only if $\pi(n_1) = fact$. Following the *join query* and the *mapping convention*, we define the query of p as a BGP q_p that has a single distinguished variable that is the distinguished variable of path's last node's mapping $HeadVars(q_p) = HeadVars(\delta(n_m))$. The body results from the combination the bodies of the queries of the path's nodes and edges $Body(\delta(n_1)), Body(\delta((n_1, n_2)), ..., Body(\delta(n_m))$.

Example 3: Consider the fact-rooted MD path (n_{Film}, $(n_{Film}, n_{Director})$, $n_{Director}$, $(n_{Director}, n_{Gender})$, n_{Gender}) in Fig 2 and the mappings in Table II. Query of this path is $q_p(?gender)$:- $?film\ \ wdt{:}P31\ \ wd{:}Q11424$, $?film\ \ wdt{:}P57\ \ ?director$, $?director\ \ wdt{:}P31\ \ wd{:}Q5$, $?director\ wdt{:}P21\ ?gender$, $?gender\ wdt{:}P31\ wd{:}Q48277$.

[9] PREFIX wdt: <http://www.wikidata.org/prop/direct/>

[10] PREFIX wd: <http://www.wikidata.org/entity/ >

[11] https://en.wikibooks.org/wiki/SPARQL/Prefixes. (Accessed 17-Sep-2017).

[12] Corresponding to *rdf:type*.

IV. Multidimensional Analysis

In this section, we introduce multidimensional analysis using SMDS. First, we present the notion of analysis situations (AS) to model and specify MD queries. Afterwards, we provide a method to automatically translate MD queries against an SMDS to SPARQL queries against the data sources.

A. Analysis Situations

We adapt the notion of analysis situation (AS) from analysis graphs [6] originally proposed for relational OLAP. An analysis situation is a high-level specification of an MD query pattern which corresponds to a specific business need or goal. This specification is expressed in terms of the MD elements involved and the way they are involved. As a consequence, constructing an MD query using analysis situation does not require out-of-the-box skills or technical details. Analysis situations are created by experts on the schema level and instantiated by analysts. Variables in an analysis situation can be bound by the analyst to concrete values meeting the analyst's needs. In principle, variables can occur anywhere in an analysis situation, including both schema and instance levels. An analysis situation can be used to freely query SMDS, i.e., without a predefined pattern. A bound analysis situation against an SMDS can be automatically translated to a SPARQL query against the data sources using the mappings.

Definition 6: An analysis situation AS against a multidimensional graph MDG is defined as $< f, MEAS, DIMS >$ where: f is the fact, i.e., $Fact(f)$. $MEAS$ is a set of elements, each is composed of an aggregation function applied to a measure, i.e., $< m_i, aggFunc_i >$ where $Measure(m_i)$ and $HasMeasure(f, m_i)$. $DIMS$ is a set of dimensions, each dimension is of the form $< d_i, DC_i, SC_i, gr_i >$:

- d_i is the selected dimension name, i.e., $Dimension(d_i)$.
- DC_i is dice specification $DC_i =< dcLvl_i, dcVal_i >$ where $Level(dcLvl_i)$, $HasLevel(f, dcLvl_i)$, and $BelongsToDimension(dcLvl_i, d_i)$.
- SC_i is slice specification $SC_i =< scPos_i, scCond_i >$ where $(BelongsToDimension\ (scPos_i, d_i)$, $Level(scPos_i)$, and $HasLevel(f, scPos_i))$ or $(Descriptor(scPos_i)$ and $\exists l$ such that $Level(l)$ and $(BelongsToDimension(l, d_i)$, $HasLevel(f, l)$, and $HasDescriptor(l, scPos_i)))$.
- gr_i is the level of granularity where $Level(gr_i)$, $BelongsToDimension(gr_i, d_i)$, and $HasLevel(f, gr_i)$.

Intuitively, the dice operation selects a subset of the data based on a single level value. The slice operation selects a subset of the data based on conditions over levels or descriptors. Rolling up changes the granularity level to a higher level in the hierarchy. Consequently, basic OLAP operations change the specification of the analysis situation at hand.

Example 4: Figure 3 shows an analysis situation for counting the number of the films and the average duration for the films published in the second half of 2017, of a specific *Genre*, and a specific *Continent* of origin, grouped by the *Country* of origin and a variable granularity level on *Direction* dimension. The dice values on *Genre ?diceValOnGenre*, on *Continent ?diceValOnContinent*, and the granularity on *Direction ?grOnDirection* are variables to be bound by the analyst to the values of interest. Note that *?grOnDirection* is a schema variable. Unbound variables are ignored, e.g., in case of not binding *?diceValOnGenre*, all genres will be considered.

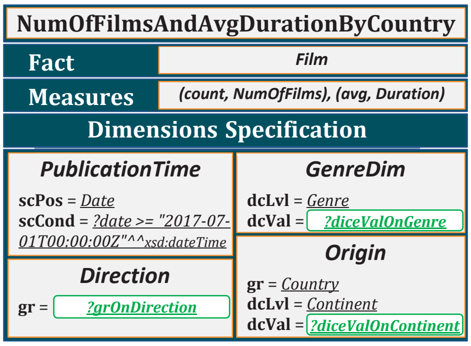


Fig. 3. Analysis situation example

B. Generation of SPARQL Aggregation Queries

In the following we describe how to transform an analysis situation to a corresponding SPARQL aggregation query. The *SPARQL aggregation queries* are of the form [14]: $SELECT\ RD\ WHERE\ GP\ GROUP\ BY\ GRP$. RD is the result description, which is a subset of the variables used in GP. GP is a BGP and could use functions such as BIND and constraints such as FILTER. GRP is a set of grouping variables. RD is composed of RD_{agg} which is aggregation functions applied to measure variables, and RD_{sel} which is a non aggregation part.

The main idea is that, starting from a bound analysis situation, a query for the dimensions q_d and a query for the measures q_m are generated[13]. Afterwards, both queries are used to generate the final aggregation query q_{out}. The head variables of the dimensional query q_d result from the granularities specified in the bound analysis situation. The body of q_d results from combining the bodies of the *queries* resulting from the *fact-rooted MD paths* to each of the MD elements involved in the dimensions' specifications (see Definition 5). Slice and dice are added as FILTER statements. The body of the measure query q_m results from combining the bodies of the queries of the fact-rooted MD paths to each of the measures.

Algorithm 1 shows the steps followed to generate q_{out}. Binding the variables is done in Line 2. The dimensional query is generated in Lines 3-14. The dimensional query body is generated in Line 4. Granularity is added to the dimensional query head (Lines 5-7). Dice and slice are added as FILTER statements to the dimensional query (Lines 8-13). The measure query is generated in Lines 15-18. The select variables RD_{sel} and the group by variables GRP are assigned from the head variables of q_d (Lines 19-22). Each head variable of a measure query is added with the aggregation function applied (Lines

[13] We use the notions of classifier (dimensional) and measure queries [3].

Algorithm 1 SPARQL generation from analysis situation

```
INPUT: $MDG$ the MD graph, $AS$ the analysis situation
OUTPUT: $q_{out}$ the resulting SPARQL query of the form SELECT RD WHERE GP GROUP BY GRP
1:  initialize($q_{out}, q_d, q_m$)
2:  $AS_{bound} \leftarrow$ BindVariablesFromUser($AS$)
3:  for all ($spec \in$ getDimensionsSpecifications($AS_{bound}$)) do
4:      Body($q_d$) $\leftarrow$ Body($q_d$), generateDimMDPathsQueryBody($MDG$, $spec$)
5:      if ($gr \neq null$) then
6:          HeadVars($q_d$) $\leftarrow$ HeadVars($q_d$) $\cup$ HeadVars($\delta(\lambda^{-1}(gr))$)
7:      end if
8:      if ($dcLvl$ $and$ $dcVal \neq null$) then
9:          addFliter(Body($q_d$), HeadVars($\delta(\lambda^{-1}(dcLvl))$), $dcVal$)
10:     end if
11:     if ($scPos$ $and$ $scCond \neq null$) then
12:         addFliter(Body($q_d$), HeadVars($\delta(\lambda^{-1}(scPos))$), $scCond$)
13:     end if
14: end for
15: for all ($agM \in$ getAggregatedMeasuresSpecification($AS_{bound}$)) do
16:     Body($q_m$) $\leftarrow$ Body($q_m$) , generateMeasureMDPathQueryBody($MDG$, getMeasure($agM$))
17:     HeadVars($q_m$) $\leftarrow$ HeadVars($q_m$) $\cup$ HeadVars($\delta(\lambda^{-1}$(getMeasure($agM$))))
18: end for
19: for all $v \in$ HeadVars($q_d$) do
20:     $RD_{sel} \leftarrow RD_{sel} \cup \{v\}$
21:     $GRP \leftarrow GRP \cup \{v\}$
22: end for
23: for all ($agM \in$ getAggregatedMeasuresSpecification($AS_{bound}$)) do
24:     $RD_{agg} \leftarrow RD_{agg} \cup$ createAggregation(HeadVars($\delta(\lambda^{-1}$(getMeasure ($agM$)))), getAggregation($agM$))
25: end for
26: $GP \leftarrow$ Body($q_d$) , Body($q_m$)
```

23-25). Finally, the bodies of the dimensional and the measure queries are combined and added to GP (Line 26).

Example 5: Listing 1 shows the SPARQL query generated from binding the analysis situation depicted in Fig. 3. *wd:Q157443* (*comedy film*) is assigned to *?diceValueOnGenre* and no value is assigned to *?diceValOnContinent*, i.e., all continents are considered. Furthermore, *Director* level is assigned to *?grOnDirection*. *Direction* dimension specification adds *?director* to RD_{sel} and to GRP (group by) and Lines 7 and 8 to the query body. *Origin* dimension specification adds *?country* to RD_{sel} and to GRP and Lines 12 and 13 to the query body. *PublicationTime* dimension specification adds Lines 9 and 16 to the query body but does not affect RD_{sel} or GRP as it does not specify a granularity. *Genre* dimension specification adds Line 10, 11, and 15 to the query body but does not affect RD_{sel} or GRP.

Listing 1. SPARQL generated for the analysis situation in Fig. 3

```
1  SELECT
2    (AVG(?duration) AS ?avgDuration)
3    (COUNT(?film) AS ?countNumOfFilms)
4    ?director ?country
5  WHERE {
6    ?film wdt:P31 wd:Q11424.
7    ?film wdt:P57 ?director.
8    ?director wdt:P31 wd:Q5.
9    ?film wdt:P577 ?date.
10   ?film wdt:P136 ?genre.
11   ?genre wdt:P31 wd:Q201658.
12   ?film wdt:P495 ?country.
13   ?country wdt:P31 wd:Q6256.
14   ?film wdt:P2047 ?duration.
15   FILTER (?genre = wd:Q157443)
16   FILTER (?date >= ”2017-07-01T00:00:00Z”^^xsd:dateTime)
17 }
18 GROUP BY ?director ?country
```

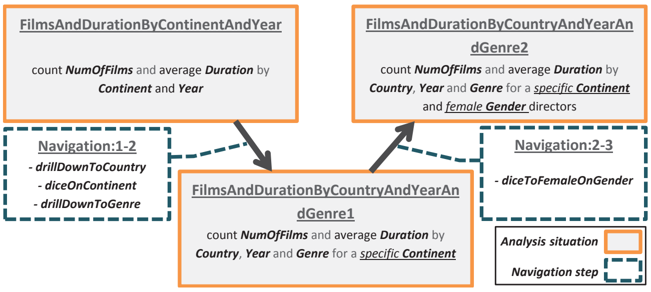

FilmsAndDurationByContinentAndYear
count NumOfFilms and average Duration by Continent and Year
FilmsAndDurationByCountryAndYearAndGenre2
count NumOfFilms and average Duration by Country, Year and Genre for a specific Continent and female Gender directors
Navigation:1-2
- drillDownToCountry
- diceOnContinent
- drillDownToGenre
FilmsAndDurationByCountryAndYearAndGenre1
count NumOfFilms and average Duration by Country, Year and Genre for a specific Continent
Navigation:2-3
- diceToFemaleOnGender
Analysis situation
Navigation step

Fig. 4. A simplified example of a semantic web analysis graph from the running scenario

C. Semantic Web Analysis Graphs

Analysis graphs [6], originally proposed for relational warehousing, “represent expert knowledge about analysis processes” which are “modelled pro-actively at schema level and used at instance level”. On top of SMDS, we extend analysis graphs to the semantic web, to become semantic web analysis graphs (SWAG). Being a model of an analytical process, a SWAG defines the succession of analysis situations in the course of the analysis, representing a best-practice procedure towards an analytical task. We refer to related work for the general characteristics of analysis graphs [6], [15], which basically apply also to RDF data analysis. The nodes of an analysis graph are analysis situations, presented and adapted to the semantic web in Sect. IV-A, while the edges of an analysis graph are navigation steps, each representing one or more OLAP operations that transform the input analysis situation into the output analysis situation. Analysis graph schema may contain variables at both analysis situations and navigation steps, which are bound during the course of analysis. Analysis graphs are a facilitator for self-service OLAP as they provide access to prepared reports [5]. Figure 4 shows an example of an undetailed analysis graph from the motivating scenario.

V. Discussion and Future Work

We provide a proof-of-concept implementation of SMDS and SWAG as a web application [16][14]. The implementation allows to set up an *OLAP endpoint* that enables analysts to analyze RDF data sources using SWAG developed on top of SMDS. The SMDS metamodel (Fig. 1) is implemented as OWL ontology which can be instantiated to express superimposed multidimensional schemas for specific cases. An ontology formalizes the analysis situation metamodel (Definition 6). Instances of the analysis situation ontology, expressing specific analysis situations, refer to instances of the SMDS ontology, expressing the underlying MD schema. Thus, SMDS and the corresponding analysis situations can be published over the web. Analysts interact with the endpoint to express MD queries either freely using an empty analysis situation or through instantiating a predefined analysis situation by binding its variables. In both cases, an auto-completion facility supports

[14] A video demonstrating the prototype implementation can be found under https://www.youtube.com/watch?v=ymhkqla8J1I&t=174s

analysts by providing possible, valid values for both schema and instance variables. The SPARQL query corresponding to the bound analysis situation is automatically generated following Algorithm 1 and sent to the dataset endpoint. Afterwards, the results of the SPARQL query are retrieved and can be viewed and manipulated in tabular form or as charts. The Jena[15] (ARQ) libraries are used to handle the ontologies and generate SPARQL queries.

Increasing amounts of RDF data are becoming available for public access, e.g., sets of LOD [2]. Such data are a significant resource of knowledge that is still largely unexploited for data analysis purposes. In order to render such data accessible to casual analysts, these data should ideally correspond to a multidimensional structure and self-service facilities should be provided on top of this structure. Whereas each of the related approaches falls short of satisfying that, the approach presented in this paper fulfills these requirements. First, the multidimensional model is a widely-accepted de facto standard for OLAP, and superimposition copes well with the nature of RDF data which potentially consist of millions of triples subject to frequent changes. Second, analysis situations allow analysts to query RDF sources in a way they are familiar with, which only involves MD concepts while being spared technical details. Furthermore, analysis situations facilitate self-service data analysis, minimizing the barriers for casual analysts towards external RDF data analysis.

Future work will employ SMDS as presented in this paper as a foundation for realizing an extended version of our proposal [17] of SWAG as only sketched in this paper. Semantic web analysis graphs will establish an elaborated self-service approach of OLAP analysis over RDF data sources and user studies will investigate the usability of the approach. Furthermore, we will provide tool support for specialists to design SMDS and SWAG as well as tool support for analysts to modify SWAG on-the-fly.

References

[1] A. Vaisman and E. Zimányi, *Data Warehouse Systems: Design and Implementation*. Springer, 2014.

[2] M. Schmachtenberg, C. Bizer, and H. Paulheim, "Adoption of the linked data best practices in different topical domains," in *ISWC 2014*, ser. LNCS, vol. 8796. Springer, 2014, pp. 245–260.

[3] D. Colazzo, F. Goasdoué, I. Manolescu, and A. Roatiş, "RDF analytics: lenses over semantic graphs," in *Proc. of the 23rd WWW*. ACM, 2014, pp. 467–478.

[4] B. Neumayr, C. G. Schuetz, and M. Schrefl, "Towards ontology-driven RDF analytics," in *ER 2015 Workshops*, ser. LNCS, vol. 9382. Springer, 2015, pp. 210–219.

[5] P. Alpar and M. Schulz, "Self-service business intelligence," *Business & Information Systems Engineering*, vol. 58, no. 2, pp. 151–155, 2016.

[6] T. Neuböck and M. Schrefl, "Modelling knowledge about data analysis processes in manufacturing," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 277–282, 2015.

[7] V. Nebot and R. Berlanga, "Building data warehouses with semantic web data," *Decision Support Systems*, vol. 52, no. 4, pp. 853–868, 2012.

[8] W3C, "The RDF data cube vocabulary," https://www.w3.org/TR/2012/WD-vocab-data-cube-20120405/.

[9] B. Kämpgen, S. O'Riain, and A. Harth, "Interacting with statistical linked data via OLAP operations," in *ESWC 2012 Satellite Events*, ser. LNCS, vol. 7540. Springer, 2015, pp. 87–101.

[10] L. Etcheverry and A. A. Vaisman, "QB4OLAP: A vocabulary for OLAP cubes on the semantic web," in *Proc. of COLD 2012*, ser. CEUR Workshop Proceedings, vol. 905. CEUR-WS.org, 2012.

[11] ——, "Efficient analytical queries on semantic web data cubes," *CoRR*, vol. abs/1703.07213, 2017. [Online]. Available: http://arxiv.org/abs/1703.07213

[12] M. Golfarelli, D. Maio, and S. Rizzi, "The dimensional fact model: A conceptual model for data warehouses," *Int. J. Cooperative Inf. Syst.*, vol. 7, no. 2-3, pp. 215–247, 1998.

[13] W3C, "SPARQL 1.1 query language," https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[14] D. Ibragimov, K. Hose, T. B. Pedersen, and E. Zimányi, "Optimizing aggregate SPARQL queries using materialized RDF views," in *ISWC 2016*, ser. LNCS, vol. 9981, 2016, pp. 341–359.

[15] C. G. Schuetz, B. Neumayr, M. Schrefl, and T. Neuböck, "Reference modeling for data analysis: The BIRD approach," *Int. J. Cooperative Inf. Syst.*, vol. 25, no. 2, pp. 1–46, 2016.

[16] M. Hilal, C. G. Schuetz, and M. Schrefl, "An OLAP endpoint for RDF data analysis using analysis graphs," in *Proc. of ISWC 2017 – Posters and Demonstrations and Industry Tracks*, ser. CEUR Workshop Proceedings, vol. 1963. CEUR-WS.org, 2017.

[17] M. Hilal, "A proposal for self-service OLAP endpoints for linked RDF datasets," in *EKAW 2016 Satellite Events*, ser. LNCS, vol. 10180. Springer, 2016, pp. 245–250.

[15] https://jena.apache.org/

Creation of interactive panoramic video for phobia treatment

Dana Horváthová, Vladimír Siládi
Department of Computer Science
Faculty of Natural Sciences, Matej Bel University
Banská Bystrica, 974 01, Slovakia
dana.horvathova@umb.sk, vladimir.siladi@umb.sk

Abstract—**This paper presents the possibilities of designing new environments for virtual reality exposure therapy (VRET), effective for treating phobias. Virtual reality (VR) tools can help immerse the patient into a stressful environment and induce phobia. If this environment is sufficiently realistic and the patient can freely rotate and move in this environment at the same time, deception of senses occurs. Our paper introduces methods of creating virtual environments by the use of interactive panoramic video. We present a novel methodology for creating interactive panoramic video, explore the ways of processing this video with High Performance Computing Centre (HPCC) tools and analyze the accessible hardware and software tools. Here, we present a series of interactive panoramic videos that can be used as treatment tool in therapy.**

Keywords—**Virtual reality, virtual environments, interactive panoramic video, treatment of phobias**

I. Introduction

The use of Virtual Reality (VR) for the treatment of phobias, VRET (Virtual Reality Exposure Therapy) is the topic of many research projects and studies in the world. One of the most recent systematic review studies has conducted a deep analysis of the condition in this area [1]. This analysis confirms the high efficacy of therapeutic procedures where the patient is embedded into the simulated environment or the situation, which represents phobias for him in the real world. As the authors state: "VRET applications have become an effective alternative that can equate the results of traditional treatments for phobias from an efficacy point of view. However, they are also tools capable of enhancing the psychological treatment field. In the coming years, there will be a significant increase in the routine use of these VRET applications in clinical contexts, but there are important challenges to overcome. The most important is the acceptance of these technologies by clinicians. [1]

Additional studies [2] and [3] discuss the trajectory of Clinical VR over the last 20 years and summarize the basic assets that VR offers for creating clinical applications. "While there is still much research needed to advance science in this area, we strongly believe that Clinical VR applications will become indispensable tools in the toolbox of psychological researchers and practitioners and will only grow in relevance and popularity in the future." [2]

Or as other authors summarize [3] "Overall, meta-analyses have indicated that VR is an efficacious tool, compares favorably to comparison conditions, and has lasting effects that generalize to the real world. However, problems have been noted including small sample sizes, lack of methodological rigor, and lack of comparison groups. "

Interesting is also the paper [4], which is based on the use of the VR in medicine, or [5], where the authors focused on the system of perception of the real world through audiovisual stimulation for the treatment of different types of phobias.

As far as technologies go, today's rendering of virtual worlds is becoming more realistic, faster and more convincing. Better and more accessible VR tools, which serve to deceive senses, wipe the boundaries between the virtual world and reality. Every year, capturing capabilities of the real world are enhanced with panoramic cameras, as well as immersive imaging thanks to advances in 3D graphics and the increase in computing power. For both therapists and patients, these technologies become more accessible and easier to use.

There are currently several treatment options for phobias that are often combined together [6]. The most frequent is pharmacotherapy and psychotherapy, where cognitive-behavioral approaches, psychosocial methods and relaxation practices are the most common. However, long-term pharmacotherapy for phobias is associated with the undesirable effect of drugs, therefore a more rational choice would be the forementioned virtual reality therapy - VRET [7]. This method of phobia treatment is not at all the youngest. The first known publication on virtual reality therapy was the work "Virtual environments and psychological disorders". Max North introduced the term VRT (Virtual Reality Therapy) for the first time and described this method of therapy. It was in the years 1992-1995 within his dissertation [8]. Following was a book published in 1996 entitled "Virtual Reality Therapy, An Innovative Paradigm" [9].

The technology of virtual reality and the possibilities of its use have since been investigated and attracted the interest of scientists in the fields of informatics, psychology and various medical disciplines.

Thanks to the idea of Dr. Jan Záskalan and material support from Nettech, Ltd., (represented by Peter Sojka), we have been dealing with the creation of virtual environments and various support tools for the treatment of phobias using virtual reality at the Department of Computer Sciences, Faculty of Natural Sciences, Matej Bel University in Banská Bystrica. We prepare materials (applications, videos, etc.) for the needs of a

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

psychotherapist to support the treatment of different types of phobias, such as fear of heights, depths (hypsofobia, acrophobia), open spaces (agoraphobia), fear of spiders (arachnophobia), reptiles (herpetophobia), fear of people and crowded places, or fear of traveling in different means of transportation, meeting and communicating with people (sociophobia), and others. Because we are not equipped with the possibility of verifying the developed materials in therapeutic practice, our efforts are directed to the development of new methods of virtual environment creation.

We found out that we are not alone in Slovakia. Several institutions come up with new ideas to use virtual reality for the benefit of man. They are used for example, to induce homely athmosphere to hospitalized children, to reduce pain or to speed up recovery after a heart attack. Very good experience is also at the UVEA Mediklinik Eye Clinic in Martin, where virtual reality is used to treat successfully amblyopia https://www.uvea.sk/sk/aktuality/item/513-nove-moznosti-liecby-tupozrakosti.

A bright example of using VR for phobias is MindBox company http://www.mindbox.help/ that focuses on creating virtual environments for the treatment of different types of phobias and our department has begun to look for the intersection of interests and possibilities.

Creating a VR environment on computers has particularly high demands on compatibility, smooth implementation as well as image quality. The software must be able to run on a desktop or laptop computer (at home of the patient or in the psychotherapist's office) however for the presentation or patient's immersion, additional VR equipment is needed [6]. Our effort is to adapt to current trends, when Android system devices are used and mobile phones or tablets clamped in a simple "head-set" on the basis of so-called "google cardboard" serve as stereoscopic glasses. This combination has currently the largest user base that provides fast and inexpensive development.

II. Methods of creating virtual environments

There are several methods to create virtual environment [10]:

A. Modeling objects or environments through computer graphics

To model 3D objects in the modeling program, one of the following modeling methods can be selected:

- Using curves: the surface of the model is made up of multiple joined curves (this method is not very much used nowadays because there are also easier modeling methods).
- Using polygons/polygon modeling: the object is constructed using polygons of its surface, which consists of vertices joined to polygons (best triangles), with the advantage of being able to build any object using a polygonal network.
- Using basic geometric shapes: objects are created by the set of operations from fundamental bodies and their modification creates complex structures.

B. Animating models of objects using the animation program

The model animation system is based on inserting keyframes into the timeline while defining which parameter is to be remembered as the key for that image. This can be for example the position of the object, the rotation, the size and in the case of more complex objects also their current shape. Consequently, each frame between the two defined is automatically calculated and represents a partial change of position.

C. Shooting static objects and scenes using 360° panorama shooting technology

The static object snapshot method summarizes the steps taken to capture 360° panoramas and process them into an interactive application for a variety of devices, including inserting them into the application environment. The photo captures a relatively accurate image of a scanned scene and is less demanding to make than to model a complex subject looking realistically. However, this is only true if the scene for photography is easily accessible. Therefore, it is convenient to use a variety of technologies in the development of environments and to choose the more advantageous for each one.

D. Filming moving objects and scenes using 360° panorama shooting technology

By filming moving objects and scenes an interactive panoramic video can be created using dedicated hardware such as a camera system (or a special camera itself) that takes 360° panorama in both the horizontal and the vertical directions. In such a video, the patient can then freely choose the direction and angle of view, possibly with the help of other VR command units to control his movement in the virtual space.

E. Combination of forementioned methods

From the point of view of individualising the approach to treating each patient, it is important that the environment in which he is embedded enables the staging of the phobic state and that it is possible to choose from a multiple alternatives corresponding to his needs. Therefore, it is convenient to use the method of combining some of the forementioned methods. For example into a shot or filmed scene insert an object model whose motion and trajectory can be predefined, or the therapist can select the number of objects displayed (e.g., spiders) depending on the patient's condition.

This article focuses specifically on filming moving objects and scenes using 360° panorama shooting technology, that can serve to treat phobias by embedding into a very real looking virtual environment.

III. Types of interactive video

The main principle of interactive panoramic video (explained later) is the adaptation of the video display to the patient or to the viewer in general, where the media manipulated by the author gradually changes into the viewer-controlled medium. The viewer can communicate directly with the video through various hardware or software tools, has direct access to the video and can change some video parameters in real time. We know some of the following types of videos that are derived from the type of interaction the viewer can perform [11]:

- HTML5 interactive video,
- IVOT - Interactive Video Object Technology,

- Branching video,
- Panoramic video, which we pay the most attention to in this article.

A. HTML5

HTML5-based video is the simplest of the types of video interactions. The essence is the insertion of text, images, or other virtual objects into a pre-created video. Simple examples of such an interaction include a video with an interactive quiz at the end.

Such a video mode is suitable for use as a learning material - online courses or additional information by simply interacting with video.

We do not need any special hardware to create an HTML5 interactive video. For reasons of universality, it is best to create MP4 video supported by most players.

Many software tools suitable for work with interactive videos are available online. The most used are RaptMedia, Wirevax, or fast-growing Storygami.

B. IVOT

Interactive Video Object Technology is characterized by the use of physical objects captured on the video. In this video, the user has the ability to interact with certain parts of the scenery.

When moving the cursor to one of the captured object types, the video displays the name or properties of that object for the viewer.

As in the previous case, no special hardware is needed to create IVOT.

The easiest and the fastest program available is a software from IVOTek company that is available online and for non-commercial purposes is provided to users for free.

C. Branching video

This type of interaction provides the user with alternative parts of the video offered in real time. For example, in the educational video, the user can, after a short introduction, choose from three video sections that can be presented onwards. These video sections contain information from three different areas of the selected lesson and the user can decide on the course of the lesson from these offered options.

As in previous cases, it is not necessary to use specialized hardware to make this type of interactive video.

D. Interactive panoramic video

The last of the video interaction options is changing the point of view of the viewer in the video. We call this video an interactive panoramic video (IPV).

IPV can be created using specialized software. Here, the situation can be modeled from virtual scenery elements or with specialized hardware such as a 360° panorama camera in both the horizontal and the vertical directions. Nowadays there are a number of modifications for camera systems with one or more cameras capable of scanning 360° panoramas.

IV. Software tools for creation and processing IPV

When working with IPV, we can use several types of software which allow us to work with scanned or otherwise created video material and achieve the desired result.

A. SW for creation and capturing IPV

It is software that can be used to model a video on a computer (such as virtual world modeling) therefore avoiding the need for specialized hardware to create IPV. After the modeling phase we must save the project in a format that suits our needs for projection of material using IPV hardware.

The second option is the use of specialized software for IPV cutting and processing. In this case it is a combination of graphical software in which we are able to model the objects, spaces and scenes in 3D, and also a software in which we can create a video that uses these virtual objects we have modeled.

A standard combination of forementioned softwares can be the modeling software Blender 3D, in which we are able to create 3D models of objects, characters or spaces and the Unity 3D software, which is mainly used in amateur or semi-professional game software creation, but we can also use it for other goals. Both of these software are freely available in versions for Windows and Mac OS.

When making our videos, we used another type of software - the Samsung Gear 360 Manager mobile application that was developed to work with the Samsung Gear 360. The application can be used either as a control of the camera itself or as a camera viewfinder. To connect to the camera, Bluetooth technology was used. The app that is available for mobile devices with Android operating system can be used also for viewing and sharing recorded videos, as well as checking Samsung Gear 360 status.

B. Software for processing IPV

Each video, whether interactive or standard, must be compiled into one unit after recording of its individual parts and stored in a format suitable for further playback. The options for such compilation, cropping and video editing are provided by a group of software programs known as nonlinear video editors. These include the simplest ones like Windows Movie Maker, editors with more options and more complex workplaces like Lightworks and also professional programs with multiple options - such as Adobe Premiere Pro.

When processing our IPV series, we have used the Action Director application and although it is not mobile-friendly, it offers enough video editing functions and effects adding possibilities to IPV. It is available for the Windows operating system and comes with a panoramic camera.

V. Used special hardware

We decided to create IPV using a special camera capable of capturing the 360° panorama and then playing IPV in the glasses for virtual reality.

A. Hardware for capturing IPV

When considering the usability of individual 360° video camera systems, it is important to take into account a number of parameters such as price, weight, camera dimensions, internal memory, memory card capacity, and last but not least battery life.

Another limiting factor that affects IPV recording is video resolution. The recording resolution of modern cameras is often so high that they exceed the needs of a regular user and his user technology. The problem is arrises with the size of the files that is produced - the higher the quality of the recording, the larger the video record file.

There are two distinct types of consumer-friendly 360° cameras: single-lens and dual-lens offerings. Single-lens cameras typically capture content with better quality, while dual-lens offerings stitch content together with less-than-stellar results on some occasions. 360fly 4K, Ricoh Theta S 360, 4 Insta360 Air and LG 360 CAM are the most widely used 360° cameras and they are able to record the material in very good resolution.

In our case, we decided to use the Samsung Gear 360 camera, which was lended to us for research purposes by Nettech company. This camera is capable of recording 360° video in a sufficient quality on a plug-in memory card. Although the camera battery lasts only slightly over two hours, its dimensions are ideal even in confined spaces.

B. Hardware for playing IPV

An important part of the hardware needed to present the IPV are the VR glasses. The purpose of these devices is to present the 3D artificial environment created without any boundaries, i.e., whichever side we are looking at, the camera focuses at it as well.

There are two ways of sending the recorded material into VR glasses:

- The first is to send a video from the console or computer to the glasses using a cable connection.
- The second is to play video from a mobile device inserted directly into VR spectacles.

In both cases, the video is either composed of two sources on one display or two separate displays.

There are a couple of lenses between the screen or the screens used to play the video and the eyes of the user. These lenses focus and shape the projected 2D image for each eye to mimic the perception of the 3D environment with the human eye.

When working with hardware for IPV presentation we used VR glasses from Samsung, specifically the Gear VR. These spectacles work on the principle of direct insertion of the mobile device into spectacles and project the image by dividing one display into two parts.

C. Technical specification of IPV

IPV quality depends on several technical parameters such as resolution, frames per second and shutter speed. All of the above video specifications will be described on the aspect ratio of the video - 16: 9.

1) Resolution of IPV

For IPV purposes, the use of the highest possible resolution quality is attractive, but the problem is with the data flow rate and the overall video data volume. We need 4 MB of data per second for a 1080p video in the MP4 format. A 5-minute long video at this resolution would mean 1.2 GB of the necessary storage space for the device.

Therefore for devices with lower memory capacity or limited computing power, it is better to use lower resolutions such as 720p or even 480p,.

2) Frames per second

Technology that uses higher number of frames is called "High Frame Rate" technology. The standard frequencies for this technology are 48, 50 and 60 frames per second. In rare cases, the number of frames per second can climb up to 120 but in conjunction with ultra-high resolution 3D projection, there is a problem with the projection of the material, which can only be handled by high-performance systems.

When recording IPV, it would be ideal to use high frame rates per second. However, we are dealing with a similar problem as with high resolution video - file size and amount of information transmitted over a short time is a problem for less efficient devices.

3) Shutter speed

When taking pictures, the exposure time is the time in which the light from the surroundings reaches the camera sensor. In practice, this means that the longer the exposure time, the more light will hit the sensor.

When recording a video, the exposure time is mechanically very different to that of the camera where a charge is built up in the camera's sensor. However, the effect is the same as for cameras.

Exposure time is measured in different fractions of a second, e.g. 1/60 means that the image will be exposed to light for 1/60 second. In video, the exposure time means the amount of time that the light will fall on a single frame of the video. As a result of exposure time, the video appears sharper with higher exposure speeds. This is caused by the minimization of motion blur in individual video frames, as for each of the pictures, the light from the surroundings is shorter.

Video recording is controlled by the so-called "180 degree shutter" rule, based on two variables - the number of frames per second and the exposure time. This rule is shaped:

*Exposure time = 1 / (2*frames per second)*

As a result, motion blur is an adequate to the number of frames per second.

D. Prerequisites for IPV creation for phobias treatment

The IPV format is directly dependent on the presentation of the surrounding area to the user. Therefore it is useful if the phobias whose triggers we are able to capture on IPV are linked to space, social contact, or situation.

These problematic spaces are in particular small rooms, which can often be connected with the excessive confinement of several people (lifts), as well as large, open spaces such as meadows, squares and others.

VI. Methods of Creating panoramic video

In the following steps we will describe the procedure to help us with IPV creation. This creation process consists of four main points:

- identification of the topic of interactive panoramic video
- recording of the material itself
- subsequent processing of material in the relevant software
- preparation for the final IPV presentation along with the testing of the functionality of all the video elements.

A. Identification of the topic

In our case, it is first necessary to consult a psychotherapist who specifies the ideas and requirements for using the video for the treatment of phobias. Most of these IPVs arise for the treatment of patients – agoraphobics, who are afraid of a certain type of space or a situation.

B. Recording of the material

The recording of the broad field of view of the camera when creating the IPV include certain specifics that are good to know in advance.

The first problem is the absence of "behind the camera" space so the user can choose to look at any location around the camera, making the author of the video susceptible to several measures that need to be met before the video is recorded.

These include, in particular, the removal of interfering or unwanted elements from the camera's field of view, which would normally be beyond the camera.

Another problem is blind spots in the camera's field of view. With the Samsung Gear VR, blind spots are located around the camera's periphery exactly between the two lenses used by the camera. In a video where the camera needs to move, it is possible to partially hide the camera operator in this blind spot.

A similar problem created by the wide field of view of the camera is the selection of a site usable in our video. For the IPV, it is necessary to find a location that meets the required 360° panoramas.

C. Processing of the recorded material

Once the IPV has been recorded, it needs to be processed using one of the many available software environments designed to work with 360° video. Most commonly used programs are usually included with the hardware for recording these videos such as the 360fly Director in conjunction with the 360fly camera, the critically acclaimed but not freely available The PowerDirector or the Gear 360 Action Director that we use. Each of these tools should be capable of basic video editing features such as video editing and effects adding.

The basic procedure for creating our video series was:

- Importing raw video - every video needs to be imported into the selected software, even if no video editing is required because of the image connection. Such raw video is divided into two separate images (each image per camera lens) within a single MP4 file.
- Image Linking - depending on the software, association of individual video images can be done automatically by the software after the import of the video, or it needs to be launched manualy. Joining these images turned out to be the most time-consuming activity, where video images of five minute length were combined for more than twenty minutes. During this process, it is not possible to work with the video on any of the following points. Action Director software joins images automatically after importing videos. Percentage information of image linking is displayed in the Media window under each imported video.
- Video clipping - as with a classic video, also with IPV, it is possible to trim individual parts of the video that are not needed in the resulting video. To cut in the software used by us, it is necessary to move the imported and connected video together by dragging it from the Media window to the video track located at the bottom of the software window. After inserting a video into a video footage, we can cut the video according to our needs.
- Video editing - most software programs such as Action Director offers basic video edits, for example transitions between parts of the video, effects adding or changing the temperature of the light. We can also add or remove audio tracks when editing a video. To open the editing window in the Gear 360 Action Director, we can select the icon of three overlapping circles on the video track bar.
- Renderings - after all the edits we have applied to the video, we now need to export the video as a whole from the video editor in a form of a file in MP4 format. This can be done in standard coding AVC or H.265 coding also referred to as High efficiency video coding (HEVC). In our case, we can export the video using the Produce button on the lower right-side window of the Action Director software. Before exporting, the quality of the file you are preparing to export needs to be chosen as well as the location where you want to save the resulting file. We start the rendering process itself by setting all aspects of the rendering process by clicking the Start button at the bottom right of the rendering window. Rendering is a time-consuming activity in video processing. In the classic rendering format we measured the performance of approximately 100MB / minute, which would mean a 10-minute rendering process for a 5-minute video. The course of this process is shown at the bottom of the Gear 360 Action Director window where we can monitor the percentage video completion as well as the current process time, the expected end time of the process, and the content intensity of the currently processed file measured in MB.

D. Presentation of the final IPV

The most important factor of the presentation of final material should be the ability of the video to deceive the senses as much as possible and provide the user with the opportunity to embrace the virtual simulated environment for the treatment of phobias. The most effective VR glasses for immersion in the environment are glasses that can help to annihilate surrounding visual disturbances.

Samsung Gear 360 devices are capable to present recorded material in various formats, but not all of them are practically usable for IPV.

The software used to project videos on Samsung mobile devices, offers video presentation capabilities in the following formats:

- 2D – is the standard 2D image projection format,
- 180 – flat panoramic view of the video that causes 360° deformation of the image to 180° panorama making it unusable for our purposes,
- 360 – is the most ideal IPV playback format that faithfully presents 360° panorama to the user. We use this format to present our IPV,
- 3D SBS/TB - the last of the offered formats are SBS (side by side) and TB (top to bottom) 3D formats offered for standard 3D videos as well as panoramic 3D videos.

VII. The Use of HPCC When Processing IPV

The duration of many of the above-mentioned time-consuming processes can be shortened by distributing the task itself among multiple machines that could perform operations simultaneously on different parts of the task.

The process on which such a distribution of calculation is most commonly used is the so-called distributed or parallel video rendering that can last long hours especially for larger videos. The most common way of distributed rendering is to split images of rendered video into small parts called buckets and to subsequently assign them to machines in a distributed system. After successful rendering of the partial tasks, all the parts are rendered together into final image.

The tools needed for distributed rendering are the system itself, which is capable of parallel computing as for example High Performance Computing Center (HPCC), and also a software that can in this system divide the job between several machines. And while such rendering is currently quite common, there are very few software solutions that can be used during such video processing.

The most famous and most usable software package is V-Ray from Chaosgroup company. It can be used separately for distributed rendering but also in collaboration with many video editing software such as 3ds Max, Maya or Blender.

As a part of our work with IPV, we have not been able to make full use of the available HPCC while rendering videos because it is currently usable for scientific purposes only in a basic configuration without specific software that we could use for distributed IPV rendering.

VIII. Approach of the project

For potential patients, therapists or regular visitors, the site http://myphobia.org/ is prepared, which synthesizes several co-operating parts of our project and provides the opportunity to register and subscribe to the support system of therapy. There is also a series of IPVs designed to treat agoraphobia, claustrophobia, athephobia, and hylofobia.

In the future, we expect that thanks to the tried and tested method of creating such videos, we will be able to support the treatment of phobias by adding other required materials on the portal and thus expanding the scope for physicians and therapists to better cover the simulation of traumatic situations.

Acknowledgment

The authors would like to thank Nettech, Ltd., which provided hardware for experiments and Dr. Ján Záskalan for his valuable comments and suggestions to improve the quality of the paper. Part of the computing was performed in the High Performance Computing Centre of the Matej Bel University in Banská Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme.

References

[1] Botella, C., Alvarez, J. F., Botella, V.G., Banos, R.M., Recent Progress in Virtual Reality Exposure Therapy for Phobias: A Systematic Review in Current Psychiatry Reports, Current Psychiatry Reports, New York : Springer Science+Business Media, May 2017.

[2] Rizzo, A., Koenig, S., Is Clinical Virtual Reality Ready for Primetime?, Neuropsychology, October 2017.

[3] Maples-Keller, J. L., Bunnell, B. E., Kim, S.-J. and Rothbaum, B. O., The Use of Virtual Reality Technology in the Treatment of Anxiety and Other Psychiatric Disorders., 25, [ed.] PubMed. 3, s.l. : Harvard Review Psychiatry, pp. 103-113, May/Jun 2017.

[4] Ortiz J.S. et. al, Realism in Audiovisual Stimuli for Phobias Treatments Through Virtual Environments, rev. Augmented Reality, Virtual Reality, and Computer Graphics, Jun 2017, pp. 188-201.

[5] Renan, A.V., Araújo, L.V., Monteiro,C.B., Da Silva, T.D. and Nunes, F.L., MoVEROffice: Virtual Reality for Upper Limbs Rehabilitation, s.l. : Virtual and Augmented Reality (SVR), 2016. pp. 160-169. XVIII Symposium on IEEE.

[6] Horváthová, D., Siládi, V., Lacková E., Phobia treatment by help of virtual reality, Poprad : Informatics 2015 Proceedings. IEEE 13th International Scientific Conference on Informatics, 2015, November 18-20. ISBN 978-1-4673-9867-1.

[7] Singh, J., Singh, J., Treatment options for the specific phobias : International Journal of Basic & Clinical Pharmacology, 3/5, s.l. 2016.

[8] North, M. M., North, S., Virtual environments and psychological disorders, 4/2, s.l. : Electronic Journal of Virtual Culture, 1994, pp. 37-42.

[9] North, M. M., North, S. M., Coble, J. R., Virtual Reality Therapy, an Innovative Paradigm, s.l. : IPI Press, 1996. ISBN 1-880930-08-0.

[10] Horváthová, D., Siládi, V., Creating virtual environments for phobia treatment., 1/6, s.l. : Open Computer Science, October 2016,. (Online) 2299-1093.

[11] Dudáš, A. Methodology of creating and processing of interactive panoramic video. Metodika tvorby a spracovania interaktívneho panoromatického videa. Banská Bystrica : Thesis, s.n., 2017.

Peripheral Devices Support for LIRKIS CAVE

Marián HUDÁK, Štefan KOREČKO, Branislav SOBOTA
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovak Republic
Marian.Hudak.2@tuke.sk, Stefan.Korecko@tuke.sk, Branislav.Sobota@tuke.sk

***Abstract*—LIRKIS CAVE, designed and built at the home institution of the authors, is an immersive virtual reality environment of unique, cylinder based construction with stereoscopic video output provided by twenty 55” 3D LCD displays. While the visualization engine of the CAVE is optimized for the unusual display arrangement, motion tracking and 3D mouse input, it originally lacked in support of other input devices. In this paper we deal with the problem of extending and optimizing the peripheral devices support by introducing a socket-based approach, which connects an external input processing application directly to a distributed virtual scene rendering application.**

Keywords—Virtual Reality; Peripherals Devices; CAVE

I. INTRODUCTION

Applications of virtual reality (VR) can be found in different areas of education, research and development. In each sector, different types of VR devices, virtual environments and user interactions are used. Probably the most immersive type of VR environment is CAVE (Cave Automatic Virtual Environment), which has a form of a room with displays instead of ordinary walls. Over the last few years, several additional peripheral input devices have been developed to provide a better control of virtual environment. Although these devices are supported by VR systems, it is often necessary to calibrate them according to the needs of a CAVE system [1]. The use of control devices is often associated with a scene response delay. When the system reacts with a noticeable delay, the user is unable to experience full immersion and is often frustrated by inappropriate functionality. Therefore, it is crucial to search for causes of such imperfections and deliver solutions that will eliminate them.

The LIRKIS CAVE [2] is a compact fully immersive VR environment with unique features such as a non-typical configuration of LCD displays and transportability. It has a basis of 2.5x2.5 m and can contain up to five persons. The CAVE had originally suffered from unsatisfactory peripheral input devices support, which was limited to a motion tracking system and a 3D mouse. This problem was first solved by implementing centralized extensions for keyboard, mouse, gamepad and other devices. Because this solution caused noticeable delays a new, distributed, version has been developed. In this paper we describe the distributed version of the extension, which provides support for multiple standard and non-standard devices while paying attention to possible connectivity and latency issues.

The rest of the paper starts with a short description of the LIRKIS CAVE in section II. Section III deals with the first, centralized, peripheral devices support. The new implementation is described in section IV and both solutions are compared in section V. Section VI deals with related work and the paper concludes with a summary of achieved results.

II. LIRKIS CAVE

The overall appearance of the CAVE is defined by a 2.5m x 2.5 m x 3m steel frame, which holds twenty 55” LCD screens. Three of them are the floor, three the ceiling and the rest (14) are the walls forming 7 sides of a decagon. Every LCD provides Full HD stereoscopic output with side by side multiplexing image. The computing system of the CAVE is a cluster of seven computers, one master and six slaves. The master computer controls data flow through network and sends global scene information to slaves. All of them communicate over a local network. Each of the slave computers is equipped with NVIDIA Quadro graphics card. The slave computers render graphical output for the LCD screens. Each slave is responsible for 3 to 4 screens (Fig.1).

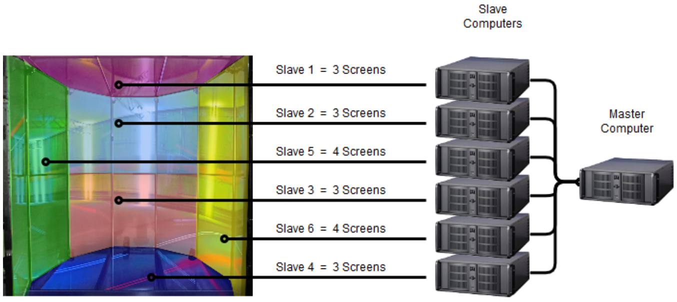


Fig. 1 LIRKIS CAVE hardware components.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

The software part of the LIRKIS CAVE contains several software modules (Fig.2). One of them is a visualization core, called Video Renderer, which creates a three-dimensional environment and renders the final scene for LCD screens. Command management and information sharing between renderers is provided by a module called Control Center. Its main task is to secure data transfer and control of I/O devices.

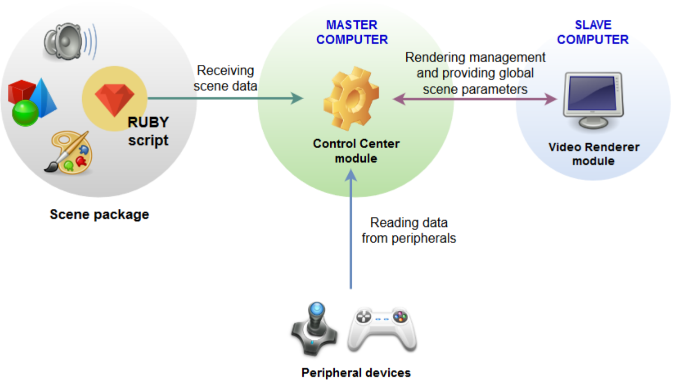


Fig. 2 LIRKIS CAVE software components and communication.

Both Video Renderer and a portion of Control Center are built on the OpenSG (http://www.opensg.org/) 3D graphics library. Virtual scenes, visualized by the CAVE are in the form of scene packages, containing audio-visual content and RUBY scripts for scene dynamics. Interaction between users and virtual environment is provided by peripheral input devices, which perform motion tracking and scene objects control.

III. Centralized Peripherals Support

Each CAVE system supports different types of peripheral devices, according to the purpose of its virtual environments. In LIRKIS CAVE the supported ones are standard input devices (mouse and keyboard) and special devices for VR environment control, such as OptiTrack, 3D mouse, Joystick, GamePad and SteeringWheel. Each device can be applied to different kinds of scenes. Typically, some devices are used to control 3D objects, while others allow the user to move around the scene [3]. Implementing the support of standard input devices doesn't cause any problem as they are natively supported by virtually all operating systems and software dealing with user interaction. This is also the case of OpenSG. On the other hand, other input devices in LIRKIS CAVE need calibration and special software for data capture and transfer. For example, motion tracking is provided by OptiTrack, which needs to be calibrated and accessed from the RUBY script of a scene package. In order to achieve a proper functioning, it is necessary to scale the input entries and use them in script scenarios.

Most CAVE systems support different input devices via separate software packages. The original LIRKIS CAVE software didn't have such a modular structure and only allowed scenes to receive and process data from the devices in RUBY scripts. The scripts had to handle all the communication with the device, including calibration. In addition, they had to be fast enough to not cause noticeable delays. This has been particularly hard to achieve in some cases as the scripts are interpreted.

Virtual CAVE systems are most often equipped with motion tracking. Similarly, LIRKIS CAVE includes a complete OptiTrack system to track a marker located on the user's 3D glasses (Fig.3). Motion tracking is supported by eight cameras located above and around the user. In case of using multiple cameras, tracking is much more accurate [4]. Each of the cameras sends motion data to the OptiTrack Motive software. Then, the information is converted into a structure containing a position of the user in the CAVE. Afterwards, the resulting values are sent to the Control Center (CC), using a socket communication. Then the CC commands Video Renderers to adjust the rendered output accordingly. Finally, the scene is adapted from the user viewpoint, which creates full immersion.

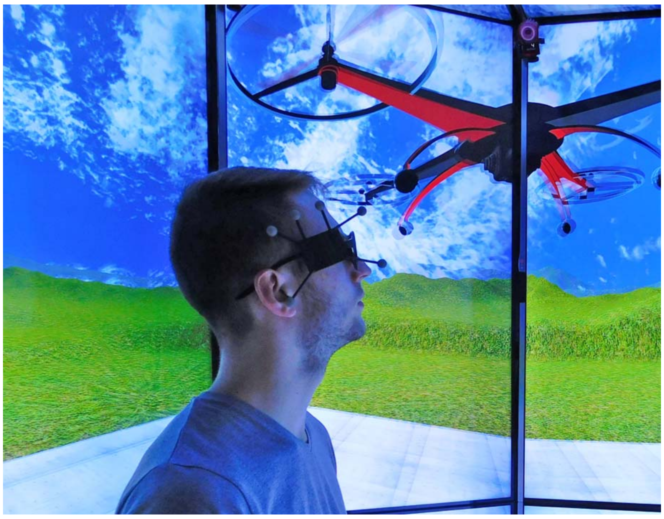


Fig. 3 User with OptiTrack marker in LIRKIS CAVE.
One of the OptiTrack cameras is visible in the corner.

The most commonly used devices in LIRKIS CAVE are JoyStick (Fig.4), Gamepad and other remote controls based on the same communication with CC. These devices use communication without time synchronization and are only sending data directly to CC. The data are sent in an infinite loop, which decreases the performance of CC. The LIRKIS CAVE does not implement event handling that can monitor device state and react correctly without delays. In the same way CC, cannot send any data to the peripherals, such as a force feedback (e.g. gamepad or joystick vibration).

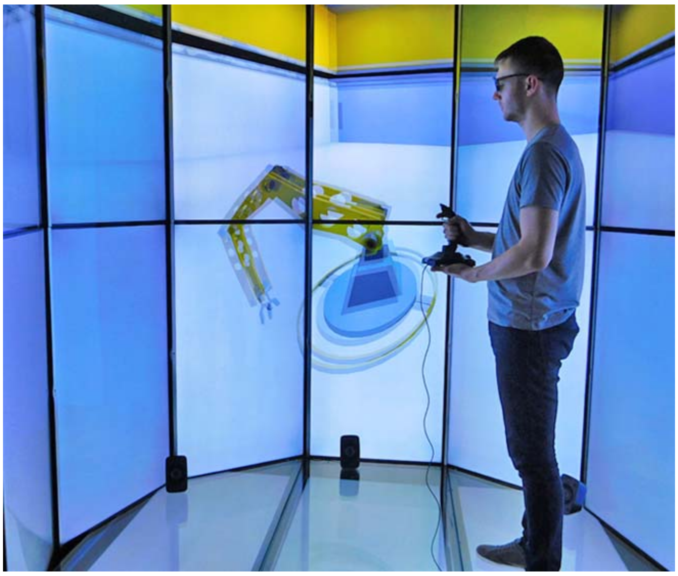


Fig. 4 Joystick used as an input device for a robotic arm control.

IV. Distributed Peripherals Support

To solve the aforementioned problems it was decided to develop an external application that will allow the devices to be interchanged without a lengthy configuration and the need of implementation on the script level. The new solution uses socket based communication and connects directly to slave computers instead of the master.

A. Socket-based communication

Sending data between system and control devices is supported by socket-based communication, which requires stable network connection [5]. One of the complications in LIRKIS CAVE was latency trough network, when the data from peripherals were replicated in loops all the time, without event handling of their state. To achieve proper behaviour and to prevent latency, it is important to choose the right type of communication between control devices and CAVE. There are two types of socket-based communication, which can be used in specific cases. The first type is a synchronous control between two communicating nodes and the second type is an asynchronous communication.

In virtual reality systems, the synchronous type of communication occurs less than the asynchronous type. There are some types of applications that require feedback or response after completing the selected activity. The most common implementation of synchronous socket communication uses requests and responses. In the LIRKIS CAVE the synchronous type is used to communicate with EEG headsets as the headsets generate data continuously and it is necessary to accumulate certain amount of data and process it before interpretation. The advantage of synchronous data transmission is their correctness and integrity.

The popularity of asynchronous communication resides in simple implementation and quick deployment, as both sides do not need to be synchronized. The principle of asynchronous communication in LIRKIS CAVE consists of opening sockets that receive information from surrounding devices connected to the same network. Data transmission is easier and also reduces system cost requirements. In this way, one can attach devices to control scene dynamics, scene motion or 3D object behavior.

B. Solution Description

The use of additional devices in the system was limited because communication with CC did not allow peripheral inputs to be expanded. The LIRKIS CAVE software had to be modified without changes to its core as its source code is not available. There were several issues to be solved: reducing system latencies when CC is receiving data, reducing the amount of computation for the master computer, improving communication between the system and input devices, and extending user interfaces with other control elements.

The reason of observed delays was simple; CC on the master computer was unable to process more data for other slave units. It was important to transfer the calculations to another part of the system. The solution was to create an external application (Fig.6) implemented in C# to connect input devices and handle data flow in the CAVE. This application is independent of the virtual cave system. It is able to control reading of data from devices, parsing them in packets and sending to sockets.

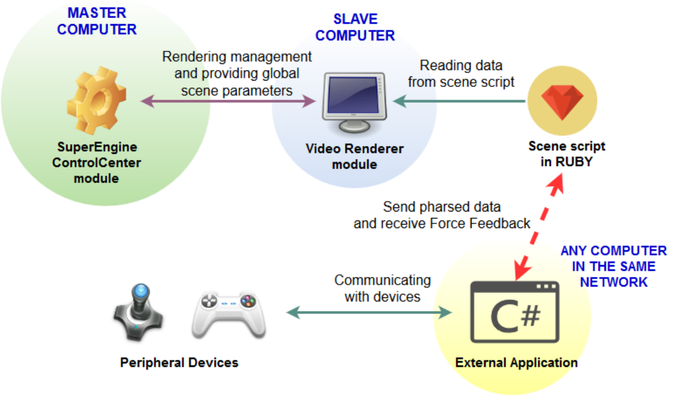

MASTER COMPUTER
SLAVE COMPUTER
Rendering management and providing global scene parameters
Reading data from scene script
SuperEngine ControlCenter module
Video Renderer module
Scene script in RUBY
Send pharsed data and receive Force Feedback
ANY COMPUTER IN THE SAME NETWORK
Communicating with devices
C#
Peripheral Devices
External Application

Fig. 6 External application to connect control devices with socket communication in LIRKIS CAVE.

Sockets are created by scene scripts and each slave computer opens its own socket for receiving data with specific port number. The external application sends data to broadcast IP address of a network subnet, where slave computers are connected. In this case, the master computer does not calculate data from control devices. All calculations are performed by the computer on which the external application is installed. The data is only received by slaves and directly attached to the scene. The external application uses event handler that sends UDP packets only if the state of the input device is changed. Sending of the UDP packets is replicated several times to ensure the transmission of information for every slave computer, but they are not sent in an infinite loop. The benefits are better communication flow over the network and lower amount of sent data. The computer with an external application must be on the same network as the virtual cave system.

The external application uses asynchronous methods, which are monitoring state of the input device and send every change to scene script. When a user starts the application (Fig.7), the main thread is always running and controls the application. Then, at least one thread is running for reading and sending data from the input device to system. When the device is disconnected, the reading thread terminates. The user can reconnect the control device or exit the application.

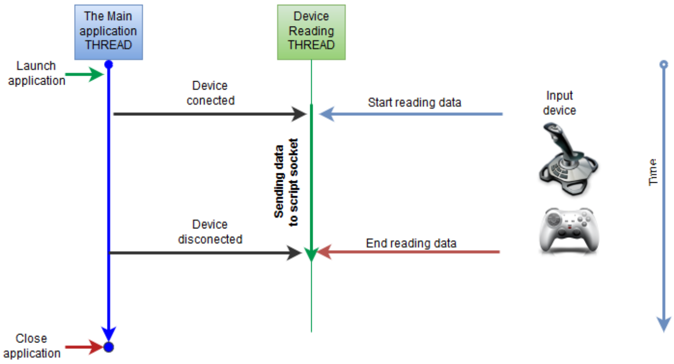


Fig. 7 Reading data from input device in external application.

The reading of data is performed by the scene script that opens the socket. When the scene is started, the main thread is executed. After the socket is created, the receiving thread is automatically started to read data from incoming packets (Fig.8). The main thread controls the scene. After data flow is terminated the scene is still running without freezing. If the packet reading is interrupted, the script waits for the next nearest one. Then the receiving thread continues its work.

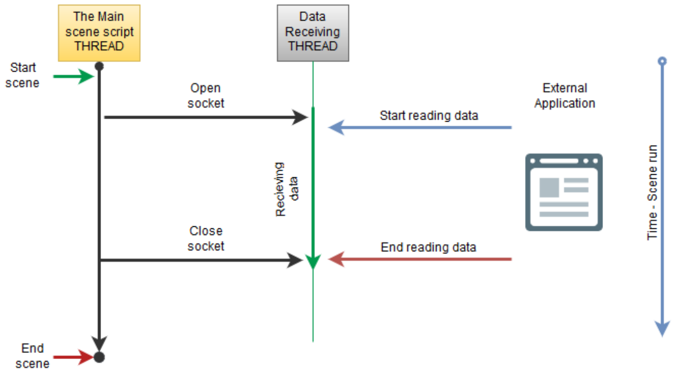


Fig. 8 Reading data by scene script with organized threads.

V. Performance Evaluation

To evaluate the new solution we measured the time needed to deliver the data from an input device to the CAVE software. For both versions the measurements were conducted in 30 replications, each consisting of 60 packet transfers. Each packet was 1024 bytes in size. The results can be seen in Fig. 8.

In the case of the previous version, the centralized solution that connects only to the master, the time needed to send data from an input device application and receive it by CC module has been measured. The average packet transmission time value was 680 milliseconds. For the new solution that connects the external application directly to the slave computers, the time from sending the packet from the external application until the packet was received by the script of the scene was measured. The measurement showed that the average packet transmission value was 410 milliseconds.

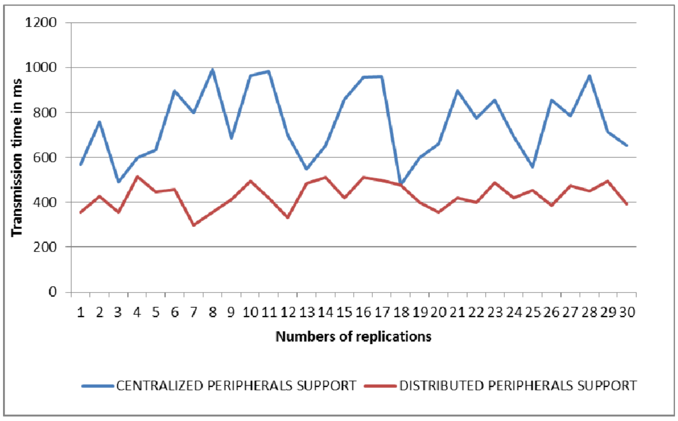


Fig. 9 Comparison of measured transmission time for both solutions.

The measurements confirmed advantages of the new, distributed, implementation, which does not require high system performance (Fig.9). This makes the system resources usable for processing needs [6]. The external application also allows the user to connect different types of control devices that can be combined and work at the same time.

VI. Related Work

Using external application for peripherals control is not exclusive for LIRKIS CAVE. Similar approach is utilized in #FIVE (Framework for Interactive Virtual Environments) [7]. #FIVE provides a communication interface and supports input control devices. In real-time communication, #FIVE is able to connect only one control device at the same time. In LIRKIS CAVE the external application creates a similar interface; however the number of simultaneously connected devices is virtually unlimited. In addition, our solution also supports smartphones, sensors, graphical interfaces and gyroscopic devices. However, #FIVE interface also includes output devices, such as screens.

The approach to input data processing in LIRKIS CAVE is also similar to IDEA (IRIS Development Environment for Applications) [8]. In IDEA, the user input is processed by a component called Control Pipe, which seems to be an integral part of the IDEA and not a separate application as in our solution. The communication in IDEA is based on simple text commands with predefined behavior. On the other hand, the external application of LIRKIS CAVE uses organized threads, which process inputs to integer values sent to the RUBY script of a loaded scene. How the script interprets the data is not limited in any way.

The external application communicates with the renderers via a local network. This concept can be further extended to a cloud, so virtual or real devices, connected to the CAVE, can be anywhere in the world. Here the experience of the team around vConnect [9] may come in very handy. The vConnect architecture supports network communication between a CAVE system and other real world devices by cloud with response time in cloud real-time data processing minimized by task scheduling for every workload.

VII. Conclusion

The utilization of virtual reality is increasing every year in various sectors, such as research, education, industry and healthcare. The expansion of Virtual Reality contributes to the development of new control devices [10]. Every new type of input device needs system support. Input devices employ different methods of use, functions and technology. All of these parameters are important for creating an interface through which they communicate with a CAVE system. The interface can be software, a communication channel and a way to send data [11]. In LIRKIS CAVE, control devices were connected to an internal interface system that could not process data from new control devices. Therefore, it was necessary to create a way to connect them all together with one control and without noticeable delays. The extension presented here creates an interface for adding different types of control devices that can be connected via USB, network (WIFI), or Bluetooth connection. Devices can be calibrated and deployed to scenes. With the new solution the system performance of LIRKIS CAVE is noticeably higher, which was also confirmed by data transmission measurements. Current LIRKIS CAVE modularity allows users to expand support for different input devices and upgrade their functionality. Thanks to the fact that the peripherals support application is external to the LIRKIS CAVE software, it can be used with other VR visualisation environments as well.

Acknowledgment

This work has been supported by the KEGA grant no. 083TUKE-4/2015 „Virtual-reality technologies in the process of handicapped person education“.

References

[1] D.R. Mestre: "Perceptual calibration in virtual reality applications." *Electronic Imaging* 2016, no. 4, 2016, pp. 1-6.

[2] M. Hudák, Š. Korečko, B. Sobota: “On Architecture and Performance of LIRKIS CAVE System“, in *8th IEEE International Conference on Cognitive Infocommunications*, Debrecen, 2017, pp. 295-300.

[3] Ch. Christou, A. Tzanavari, K. Herakleous, Ch.Poullis. "Navigation in virtual reality: comparison of gaze-directed and pointing motion control." In *Electrotechnical Conference (MELECON), 2016 18th Mediterranean*, 2016, pp. 1-6.

[4] M. Piszczek: "Positioning of objects for real-time application of virtual reality." *Przegląd Elektrotechniczny* 92, no. 11, 2016, pp.79-82.

[5] P.George: "Evaluation of Smartphone-based interaction techniques in a CAVE in the context of immersive digital project review." *IS&T/SPIE Electronic Imaging, The Engineering Reality of Virtual Reality 2014.* Vol. 9012. SPIE, 2014.

[6] Lv. Fang, et al.: "Dynamic I/O-aware scheduling for batch-mode applications on chip multiprocessor systems of cluster platforms." *Journal of Computer Science and Technology* 29.1 (2014): 21-37.

[7] R. Bouville, V. Gouranton, T. Boggini, F. Nouviale, B. Arnaldi: "# FIVE: High-level components for developing collaborative and interactive virtual environments." In *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2015 IEEE 8th Workshop on*, 2015, pp. 33-40.

[8] W.N. Griffin, et al.: “Application creation for an immersive virtual measurement and analysis laboratory.” In: *2016 IEEE 9th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, Greenville, SC, pp. 1–7.

[9] Y. He., Z. Zhang, X. Nan, N. Zhang, F. Guo, E. Rosales, L. Guan: *“vConnect: perceive and interact with real world from cave”.* Multimedia Tools Appl. **76**(1), 2017, pp.1479–1508

[10] B. Sobota, Š. Korečko, P. Pastornický and L. Jacho: Education Process and Virtual Reality Technologies. In: ITRO - in *A journal for information technology, education development and teaching methods of technical and natural sciences*., vol. 1, no. 1, 2016, p. 286-291.

[11] W. Stuerzlinger, P. Andriy, N. Dayson: "TIVS: temporary immersive virtual environment at simon fraser university: a non-permanent CAVE." In *Everyday Virtual Reality (WEVR),* 2015, pp. 23-28.

Development of Oberon-0 using YAJCo

Sergej Chodarev and Michaela Bačíková
Department of Computers and Informatics
Technical University of Košice, Slovakia
{sergej.chodarev, michaela.bacikova}@tuke.sk

***Abstract*—YAJCo is a tool for the development of software languages based on an annotated language model. The model is represented by Java classes with annotations defining its mapping to concrete syntax. This approach to language definition enables the abstract syntax to be central point of the development process, instead of concrete syntax. In this paper a case study of Oberon-0 programming language development is presented. The study is based on the LTDA Tool Challenge and showcases details of abstract and concrete syntax definition using YAJCo, as well as implementation of name resolution, type checking, model transformation and code generation.**

***Index Terms*—Abstract syntax, experience report, language development, Oberon-0, parser generator, YAJCo**

I. Introduction

Development of computer languages, especially domain-specific languages (DSL) [1], is an active research topic. Several tools are developed that aim to support the implementation of language processors and their components including parsers, type checkers, code generators and also editing tools [2]. To compare different language development tools and formalisms, the Language Descriptions, Tools and Applications (LTDA) community defined a tool challenge[1]. Results of the challenge were later published in a special issue of the *Science of Computer Programming* journal [3].

The challenge lied in the development of a compiler for the *Oberon-0* language. *Oberon-0* is very simple general-purpose programming language similar to Pascal. It was defined by Niklaus Wirth in his *Compiler Construction* book [4] as a subset of the Oberon programming language that is simple, but at the same time contains most of the important features of general-purpose programming languages.

In this paper a partial solution of the challenge implemented using the YAJCo language processor generator [5] is presented. Main contributions of this case study are the following:

1) Implementation of the LDTA Tool Challenge using YAJCo allows to compare YAJCo with other language development tools based on the already published solutions of the same challenge [3].
2) The case study allows to demonstrate the approach of language development based on the abstract syntax definition using an object-oriented language and discuss challanges connected with this approach.

II. YAJCo

YAJCo[2] (Yet Another Java Compiler Compiler) is an annotation based parser generator [5] — it allows to specify language syntax using annotated Java classes and generate a parser that would create instances of these classes based on the parsed sentence.

Definition of the language is derived from the classes that correspond to the abstract syntax of the language. Each class represents a language concept and corresponds to a non-terminal symbol in the grammar definition. Relations between classes (inheritance and composition) are used to construct the right-hand side of the grammar rules. Information that cannot be extracted from the classes, for example details of concrete syntax, are provided in form of Java annotations.

The generated parser is able to process text according to the language definition and create interconnected instances of the language concept classes. The composition relations between instances define a tree corresponding to the abstract syntax tree (AST). YAJCo can also automatically resolve references turning the tree into a graph.

In addition to the parser, YAJCo can generate *a visitor* — an abstract class implementing depth-first tree traversal according to the visitor design pattern [6]. Based on the visitor, YAJCo generates *a pretty-printer*, that transforms the object graph back into the textual form.

III. Implementation of Oberon-0

The Tool Challenge defines several problems to solve that can be combined in different ways. There are five compiler development tasks: (T1) parsing and pretty printing, (T2) name binding, (T3) type checking, (T4) source-to-source transformation and (T5) code generation. These tasks can be solved for four language levels. The basic level L1 defines a subset of the Oberon-0 without procedures and with only primitive types, L2 adds *for* loop and *case* statement, L3 adds procedures, and L4 adds composite data types.

For this paper all tasks (T1–5) were implemented for the first two language levels (L1–2). The implementation of the language[3] consists of the following components:

1) Metamodel — language abstract syntax definition in form of annotated Java classes.

[1] The LTDA'2011 Tool Challenge is described at http://ldta.info/tool.html

[2] Available at https://github.com/kpi-tuke/yajco

[3] Source code of the implementation can be downloaded at https://git.kpi.fei.tuke.sk/sergej.chodarev/yajco-oberon0

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

2) Model analysis and transformation modules: name resolver, type checker, transformer and code generator.

YAJCo is used to generate parser, pretty-printer and visitor from the metamodel. The visitor is then extended by all components that implement the model analysis and transformation, because they need to traverse the AST for their functionality.

The development process was based on test-driven development [7]. The metamodel was defined incrementally by adding more language constructs in each step and immediately testing the generated parser. Small Oberon-0 programs were defined as test cases to ensure correct processing of the language constructs. The analysis and transformation modules were also implemented incrementally with added language concepts.

IV. Syntax definition

Diagram of classes that form the abstract syntax of the base level L1 of the Oberon-0 language is presented in Figure 1. The root of the language is a *Module* concept. *Module* contains different kinds of declarations and a sequence of statements. Declarations and statements may contain expressions that include references to variables or constants and various binary and unary operators, such as addition (+) or equality testing (=).

All concepts of the language are represented by Java classes. We recognize three types of relations between concepts:

1) Inheritance (is-a) — relation between an abstract concept and more concrete concepts. For example, *TypeDeclaration* is a concrete type of *Declaration*.
2) Composition (has-a) — relation between a language concept and its parts. For example, an *Assignment* contains an *Expression* used to compute assigned value.
3) Reference — relation where a concept does not directly contain another concept, but only a reference to it. For example, an *Assignment* contains a reference to a *Variable* defined in the *Declarations* section.

Most of the concepts defined in the abstract syntax correspond directly to non-terminals in the grammar definition of the Oberon-0 [4]. Abstract syntax, however, does not follow concrete syntax definition in all details. First of all, abstract syntax contains explicit references to other concepts, instead of referencing them implicitly using identifiers. It also omits non-terminals that represent only a detail of concrete notation, for example *IdentList* used to group identifiers with the same type.

On the other hand, the abstract syntax contains concepts that are not directly present in the grammar, like *PrimitiveType* or *Boolean*. Primitive type names (`BOOLEAN` and `INTEGER`) and boolean values (`TRUE` and `FALSE`) are defined in Oberon-0 as built-in identifiers instead of keywords, so they are not included in the grammar. They are, however, concepts of the language, so they are part of the abstract syntax definition.

Definition of the abstract syntax using Java classes is a straightforward process, so rest of this section is devoted to description of the concrete syntax definition added to these classes.

A. Concrete syntax definition

Listing 1 provides an example of the language concept definition. It defines the *Module* concept with two allowed concrete syntax forms: with statements and without them (a corresponding grammar fragment in the extended BNF is provided in Listing 2). Each syntax form is defined using the class constructor and composition relations between classes are derived from the types of constructor parameters.

Keywords that are used in the language concept concrete syntax are defined using *@Before* and *@After* annotations. Notice that additional syntax constraints can be checked inside the constructor body, for example, Oberon-0 requires module name to be repeated at the end of the definition after the `END` keyword.

Listing 1. Definition of the Module concept

```
public class Module {
    private String name;
    private Declarations declarations;
    private StatementSequence statements;

    @Before("MODULE") @After(".")
    public Module(
            @After(";") String name,
            Declarations declarations,
            @Token("name") @Before("END")
            String nameRepeated) {
        if (!name.equals(nameRepeated))
            throw new RuntimeException("...");
        this.name = name;
        this.declarations = declarations;
    }

    @Before("MODULE") @After(".")
    public Module(
            @After(";") String name,
            Declarations declarations,
            @Before("BEGIN")
            StatementSequence statements,
            @Token("name") @Before("END")
            String nameRepeated) {
        this(name, declarations, nameRepeated);
        this.statements = statements;
    }
    ...
```

Listing 2. Definition of the module in the EBNF form

```
module = "MODULE" ident ";" declarations
       ["BEGIN" StatementSequence] "END" ident "." .
```

YAJCo also provides special support for most common syntactic patterns, for example sequences with separator symbols or infix operators. Using the *@Operator* annotation it is possible to define priority level and associativity of the operator as shown in Listing 3.

To support enclosing operators in parentheses, the base class *Expression* is marked with the *@Parentheses* annotation. YAJCo uses this information to automatically generate the corresponding grammar rules without the need to modify the metamodel structure to support these concrete syntax features.

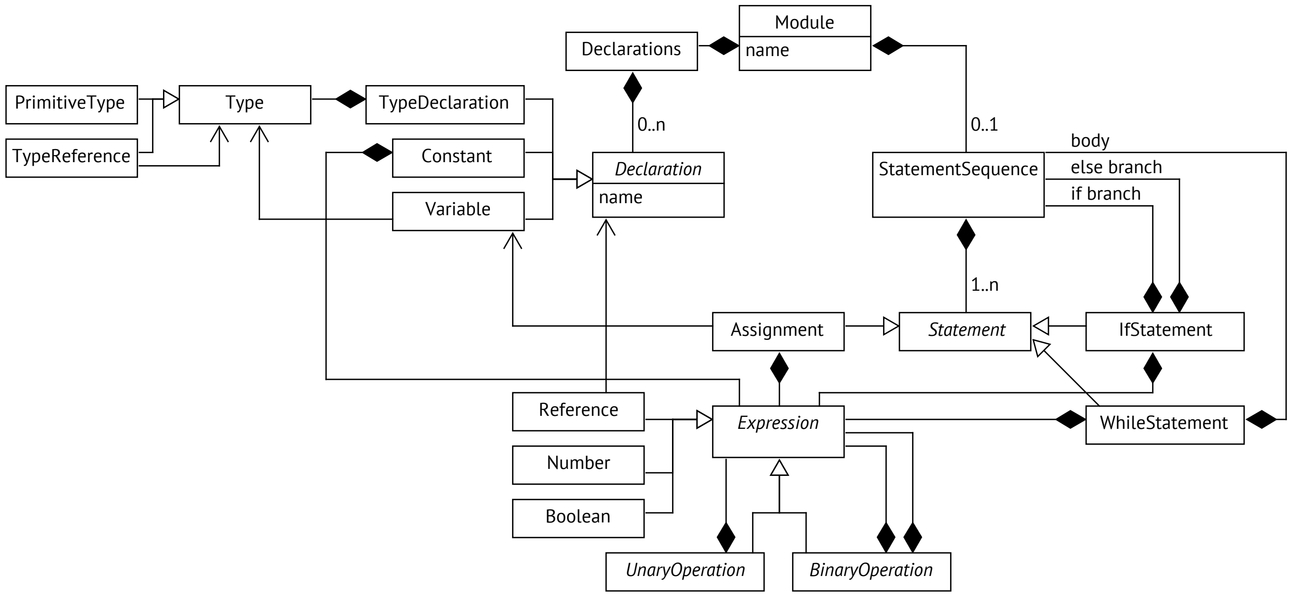


Fig. 1. Metamodel representing the abstract syntax of the Oberon-0 L1 (concrete operators are omitted)

Listing 3. Definition of the addition operator

```
public class Add extends BinaryOperation {
    @Operator(priority = 2, associativity = LEFT)
    public Add(Expression left,
               @Before(″+″) Expression right) {
        super(left, right);
    }

    @Override
    public Type getType() { return INTEGER; }
}
```

B. Helper classes for concrete syntax mapping

In some cases the relation between the abstract and concrete syntax is more complex. For example, the Oberon-0 grammar defines an ELSIF keyword used as a part of the *if* statement. It is actually a shortcut that allows a more convenient nesting of the *if* statement inside the *else* branch of the previous statement. As this construct is just “syntactic sugar” it is not required to represent it in the abstract syntax graph using a special kind of object.

In such cases special classes need to be introduced to represent concrete syntax features instead of abstract syntax concepts. In case of the ELSIF keyword, an *ElsifFragment* class was defined with the syntax definition of this element. In the corresponding constructor of the *IfStatement* class, the nested *if* statement is extracted and stored as a single statement of the *else* branch (see Listing 4). The *ElsifFragment* class, therefore, influences the grammar and parser of the language, but its instances are not stored in the abstract syntax graph.

Listing 4. One of constructors of the *IfStatemetns* class

```
public class IfStatement extends Statement {
    ...
    @Before(″IF″) @After(″END″)
    public IfStatement(
            Expression condition,
            @Before(″THEN″)
            StatementSequence thenBranch,
            ElsifFragment elsif) {
        this.condition = condition;
        this.thenBranch = thenBranch;
        this.elseBranch = StatementSequence.of(
            elsif.getStatement());
    }
    ...
}
```

Another example of classes specific to concrete syntax is provided by the declarations section of the module definition. From the point of view of the abstract syntax model, *Declarations* class is a simple collection of *Declaration* objects, where *Declaration* is an abstract base class for all kinds of declarations (see class diagram in Fig. 1). In the concrete syntax, however, declarations contain separate lists of constants, types and variables (see grammar rule in Listing 5). In addition, multiple variables can be grouped together into a single declaration if they are of the same type.

Listing 5. Concrete syntax of Oberon-0 declarations (without procedures)

```
declarations = [″CONST″ {ident ″=″ expression ″;″}]
               [″TYPE″ {ident ″=″ type ″;″}]
               [″VAR″ {IdentList ″:″ type ″;″}].
```

To map the abstract syntax to the concrete syntax, several helper classes were used. Each list from the declarations section is represented using a class that is actually a list with custom constructors (see Listing 6 for an example). Details of the declaration syntax are specified in a concrete list item class (e.g. *Constant*).

Listing 6. Definition of constant declarations section

```
public class ConstantDeclarations
                    extends ArrayList<Constant> {
    public ConstantDeclarations() {}

    @Before("CONST")
    public ConstantDeclarations(
            @Range(minOccurs = 1)
            List<Constant> declarations) {
        addAll(declarations);
    }

    public List<Constant> getDeclarations() {
        return this;
    }
}
```

For variables grouped by their type, an additional helper class was defined — *VariablesGroup* with a constructor that defines the concrete syntax of the grouping. Constructor of the *VariableDeclarations* class then merges these groups into a single list.

Finally, the constructor of the *Declarations* class merges all lists of declarations into a single hash map, where they are accessible by name. Only this representation is actually part of the abstract syntax. Instances of the helper classes are not needed for further processing so they do not need to be stored.

C. Pretty-printer

Pretty-printer generator uses the same grammar extracted from the class structure and constructors. In addition, it requires *get methods* for all constructor parameters to be implemented. This means that the objects must be able to return values for these parameters even if they do not directly correspond to properties of a class. For example, you can see such get method also in Listing 6, where it is needed only for the pretty-printer.

The need to have these get methods requires some additional work in cases were concrete syntax does not match directly to the abstract syntax and helper classes are used. If their instances are not stored in the model, they should be recreated in the get methods using an inverse transformation. For example, in case of the *Declarations* class, separate get methods do filtering of the declarations list based on the declaration's type.

V. Model analysis and transformation

All model analysis and transformation operations were implemented based on the *Visitor* class generated by YAJCo. The *Visitor* class implements the visitor design pattern for the classes that are part of the syntax definition and provides default implementations for all methods allowing full depth-first traversal through the abstract syntax tree, so only methods that would do some operations need to be overridden. In addition, all of the *visit* methods of the class accept an additional parameter allowing to pass some context.

A. Name analysis

YAJCo provides its own mechanism for name resolution [8]. However, because of the limitations of its current implementation (it does not support inheritance hierarchy of referenced classes), it was required to implement a custom name resolution module.

The *NameResolver* class extends the visitor and passes the *Declarations* object that plays role of the symbol table, as a context between visit methods. As can be seen in Listing 7, referenced names are first looked up in the list of built-in constants (like `TRUE` and `FALSE`) and then in the symbol table. Found declaration is stored in the field of the corresponding reference object. In case the declaration is missing, an error message is added into a list of errors that would be reported to the user.

Listing 7. Fragment of the *NameResolver* class

```
@Override
protected void visitReference(Reference reference,
        Declarations declarations) {
    String name = reference.getName();
    Constant constant = checkBuiltinConstants(name);
    if (constant != null) {
        reference.setDeclaration(constant);
        return;
    }
    Declaration declaration =
        declarations.getDeclaration(name);
    if (declaration == null) {
        errors.add(...);
    }
    reference.setDeclaration(declaration);
}
```

The *NameResolver* also resolves variable names in assignment statements and type names in variable declarations.

B. Type checking

The type of expressions and declarations is defined as a part of the language model. Therefore, each language concept with a type provides a *getType()* method (see for example Listing 3). Oberon-0 does not have any polymorphic operations, so definition of the *getType()* methods was straightforward — in case of operators they just returned a constant value based on the type of the operator, for a variable it returns the defined type of the variable, and for a constant it returns the type of the initialization expression bound to the constant.

The type checker, therefore, just checks types of operands in expressions and control structures (for example, the condition of the *if* statement must be of type *boolean*). It also checks if in variable assignments, the type of the expression matches the type of the variable. Found type errors are collected into a list representing the result of the type checking.

C. Source-to-source transformation

Source-to-source transformation was implemented for "desugaring" of the *case* and *for* statements added in the second language level L2 (see section VI). During the processing they were transformed into the concepts of the base language level L1.

The transformation was again done using the visitor pattern. In this case the visitor analyze all instances of the *StatementSequence* class found in the abstract syntax tree. If any of the transformed statement types was found, a transformation method is applied to replace them with corresponding concepts from the L1 language level. The transformation methods just create the instances of the replacement objects and initialize them according to the translation schema.

D. Code generation

The code generation module realizes translation of the Oberon-0 into ANSI C. The languages provide similar features, so the translation is implemented using a simple visitor with a *PrintWriter* instance as a context object used to emit the generated code.

VI. Language extension

The language extension L2 according to the Tool Challenge consists of adding the new types of statements: a Pascal-style *for* loop and a Pascal-style *case* statement that where not defined in the original Oberon-0. YAJCo has a full support for language composition [9], so the L2 extension can be implemented straightforwardly by adding classes representing the new types of statements and extending the L1 *Statement* class. This addition was automatically detected by YAJCo and new concepts were incorporated into the language definition, resulting in an extended parser, pretty-printer and visitor.

In addition, the type checker was also extended, because new statements have custom type checking rules. This was done again using the usual object-oriented extension and overriding the necessary *visit* methods. The last required change was the addition of the transformation class described in Section V-C.

The name resolution class *NameResolver* did not require modifications because no additional name resolution rules were defined. The same is true for the code generation class *CCodeGenerator*, because new language constructs are removed from the graph in the source-to-source transformation step.

VII. Observations

YAJCo allows to define languages using usual object-oriented concepts of classes, inheritance, compositions and association. The classes are written directly by the language developer and not generated from the grammar definition. This means that the developer may choose the best representation of the abstract syntax for the processing. Details of the concrete syntax can be described using annotations or in more complex cases using helper classes as was described in Section IV-B.

On the other hand, helper classes and their transformation into the abstract syntax classes complicate the language definition. In the future it is desirable to identify and support more common transformations similarly to the currently supported lists and operators.

An important advantage of expressing language concepts directly using Java classes lies in the native support of integrated development environments (IDE). It is possible to use rich refactoring and code generation features provided by modern IDEs to modify and extend the language definition and all processing code.

Implementation of model analysis and transformation using object-oriented code also allows to apply usual code abstraction and modularization techniques, such as introduction of methods for repeating code and extension using inheritance. The generated visitor allows to implement tree traversal in a very straightforward and concise way. It also supports language extension, because addition or modification of language concepts that are not directly processed by the visitor subclasses would not alter their functionality — the processor would simply inherit the default implementation of the corresponding *visit* methods from the base class.

This case study also exposed some deficiencies of the current implementation of the YAJCo tool. For example, optional constructor arguments are not directly supported. The same effect can be achieved using multiple constructors, however, direct support of the *Optional* type[4] from Java 8 would be useful.

As mentioned earlier, the built-in name resolution mechanism does not support referencing different subclasses of a base class. An important missing feature is the ability to map objects from the abstract syntax graph to source code positions for better error reporting.

VIII. Related work

Domain-specific languages are successfully used in different areas, for example application logging [10], acceptance tests [11], graphic shape description [12], expert systems [13], text analysis [14], or automatic assessment of students [15]. Source code annotations can also be considered a DSLs [16]. It was shown that DSLs improve program comprehension compared to general-purpose languages [17]. Therefore tools and methods for the development of DSLs are an active research topics.

Several other language development tools were used to solve the tasks of the LDTA Tool Challenge. Part of them use attribute grammars as a formalism allowing analysis and transformation of AST, including Silver [18] and CoCoCo [19] that use a functional language for the definition of attributes.

JustAdd [20] uses object-oriented attribute grammars, where abstract syntax is represented by classes, but parser specification and its mapping to the abstract syntax is defined separately using the Beaver parser generator. In our case the Beaver can also be used as an underlying parser generator, but the specification for it is generated automatically.

Kiama [21] is an another tool based on attributed grammars. It uses Scala type classes for representing the abstract syntax and several DSLs embedded in Scala to define other properties of a language.

Simpl DSL toolkit [22] is also based on Scala and uses the case classes. They are, however, generated automatically based

[4] See https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html

on the grammar specification. Analysis and transformation of the AST is handled by a usual Scala code. In this sense this tool is similar to YAJCo with an important difference in the fact that Simpl uses concrete syntax specification as the main artifact instead of the abstract syntax.

Rascal [23] is a specialized language for meta-programming. It provides built-in constructs for syntax definition, and strategic traversal and rewriting of trees. The challenge was also solved using the Objective Caml programming language [24] that by default provides tools sufficient for language processing, including parser combinators and "type-driven" transformers.

IX. Conclusion

The paper presents an experience report of implementing the subset of the Oberon-0 language using the YAJCo parser generator. This tool supports the development driven by the abstract syntax definition instead of the concrete syntax. Together with the use of the usual object-oriented programming approach and Java language it provides possibilities to utilize the well-known software engineering tools and approaches also for language development and make this area more approachable for developers.

We plan to use the experience from this experiment to eliminate the identified deficiencies of the YAJCo tool. The design and implementation of additional patterns for concrete syntax specification is the main goal of our future research in this area. For example, support for the optional parts of a language concept could be easily added. Grouping of elements with the same properties is also a common pattern, that can be supported directly using a special annotation.

Another area of improvement is a tool support. We plan to support generation of a complete developed environment from the language definition.

Acknowledgment

This work was supported by projects KEGA 047TUKE-4/2016 "Integrating software processes into the teaching of programming" and FEI TUKE Grant no. FEI-2015-23 "Pattern based domain-specific language development".

References

[1] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Computing Surveys*, vol. 37, no. 4, pp. 316–344, dec 2005.

[2] S. Erdweg, T. van der Storm, M. Völter *et al.*, "Evaluating and comparing language workbenches," *Computer Languages, Systems & Structures*, vol. 44, pp. 24–47, dec 2015.

[3] M. van den Brand, "Introduction—the LDTA tool challenge," *Science of Computer Programming*, vol. 114, pp. 1–6, dec 2015.

[4] N. Wirth, *Compiler construction.* Addison-Wesley, 1996.

[5] J. Porubän, M. Forgáč, M. Sabo, and M. Běhálek, "Annotation based parser generator," *Computer Science and Information Systems (ComSIS)*, vol. 7, no. 2, pp. 291–307, 2010.

[6] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[7] K. Beck, *Test-Driven Development: By Example.* Addison-Wesley, 2003.

[8] D. Lakatoš, J. Porubän, and M. Bačíková, "Declarative specification of references in DSLs," in *Federated Conference on Computer Science and Information Systems (FedCSIS).* IEEE, 2013, pp. 1527–1534.

[9] S. Chodarev, D. Lakatoš, J. Porubän, and J. Kollár, "Abstract syntax driven approach for language composition," *Open Computer Science*, vol. 4, no. 3, p. 160, jan 2014.

[10] S. Zawoad, M. Mernik, and R. Hasan, "Towards building a forensics aware language for secure logging," *Computer Science and Information Systems (ComSIS)*, vol. 11, no. 4, pp. 1291–1314, 2014.

[11] T. Straszak and M. Śmialek, "Model-driven acceptance test automation based on use cases," *Computer Science and Information Systems (ComSIS)*, vol. 12, no. 2, pp. 707–728, 2015.

[12] J. Kollár and M. Spišiak, "Direction vector grammar," in *2015 IEEE 13th International Scientific Conference on Informatics.* IEEE, nov 2015, pp. 151–155.

[13] M. Woźniak, D. Polap, C. Napoli, and E. Tramontana, "Graphic object feature extraction system based on Cuckoo Search Algorithm," *Expert Systems with Applications*, vol. 66, pp. 20–31, dec 2016.

[14] M. Sičák and J. Kollár, "Supercombinator set construction from a context-free representation of text," in *Federated Conference on Computer Science and Information Systems (FedCSIS 2016)*, oct 2016, pp. 503–512.

[15] E. Pietriková, J. Juhár, and J. Šťastná, "Towards automated assessment in game-creative programming courses," in *International Conference on Emerging eLearning Technologies and Applications (ICETA 2015).* IEEE, nov 2015, pp. 1–6.

[16] M. Nosáľ, M. Sulír, and J. Juhár, "Language composition using source code annotations," *Computer Science and Information Systems (ComSIS)*, vol. 13, no. 3, pp. 707–729, 2016.

[17] T. Kosar, N. Oliveira, M. Mernik, V. J. M. Pereira, M. Črepinšek, D. Da Cruz, and R. P. Henriques, "Comparing general-purpose and domain-specific languages: An empirical study," *Computer Science and Information Systems (ComSIS)*, vol. 7, no. 2, pp. 247–264, 2010.

[18] T. Kaminski and E. Van Wyk, "A modular specification of Oberon0 using the Silver attribute grammar system," *Science of Computer Programming*, vol. 114, pp. 33–44, 2015.

[19] M. Viera and S. D. Swierstra, "Compositional compiler construction: Oberon0," *Science of Computer Programming*, vol. 114, pp. 45–56, 2015.

[20] N. Fors and G. Hedin, "A JastAdd implementation of Oberon-0," *Science of Computer Programming*, vol. 114, pp. 74–84, 2015.

[21] A. M. Sloane and M. Roberts, "Oberon-0 in Kiama," *Science of Computer Programming*, vol. 114, pp. 20–32, 2015.

[22] M. Freudenthal, "Simpl DSL toolkit," *Science of Computer Programming*, vol. 114, pp. 85–91, dec 2015.

[23] B. Basten, J. Van Den Bos, M. Hills, P. Klint, A. Lankamp, B. Lisser, A. Van Der Ploeg, T. Van Der Storm, and J. Vinju, "Modular language implementation in Rascal - Experience report," *Science of Computer Programming*, vol. 114, pp. 7–19, 2015.

[24] D. Boulytchev, "Combinators and type-driven transformers in Objective Caml," *Science of Computer Programming*, vol. 114, no. 13, pp. 57–73, 2015.

Defining Camera-Based Traffic Scenarios and Use Cases for the Visually Impaired by means of Expert Interviews

Judith Jakob*†, Kordula Kugele† and József Tick‡
*Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Budapest, Hungary
†Faculty of Computer Science, Furtwangen University of Applied Sciences, Furtwangen, Germany
‡Institute of Applied Informatics, John von Neumann Faculty of Informatics, Óbuda University, Budapest, Hungary

Abstract—**In order to develop a concept for transferring algorithms from camera-based Advanced Driver Assistance Systems to the domain of visually impaired road users, we gather the requirements of this group. On this account, we combine procedures from Software Engineering, especially Requirements Engineering, with methods from qualitative Social Sciences, namely Expert Interviews by means of Problem-Centered Interview methodology. The evaluation of the interviews results in a total of six Traffic Scenarios clustered into three categories: Orientation, Pedestrian and Public Transport Scenarios. From each Scenario, we derive Use Cases based on computer vision and state which of them are addressed in Advanced Driver Assistance Systems so that we can examine them further in the future to formulate the transfer concept.**

I. Introduction

According to the World Health Organization (WHO) [1], 285 million people are estimated to be visually impaired. While this number has decreased since the early estimates in the 1990s because of a reduction in visual impairment from infectious diseases, there is nowadays a higher risk of age-related vision loss due to an aging society.

In addition to a decrease in life quality and independence, visual impairment causes a decline in the personal mobility [2]. To counteract this, we transfer computer vision algorithms from Advanced Driver Assistance Systems (ADAS) to the domain of visually impaired road users so that the visually impaired can benefit from latest and future advancements in the large research field of ADAS. However, because the algorithms cannot be transferred without any changes, a transfer concept is necessary. In [3], we motivate the need for a transfer concept, give an overview of the current research status in camera-based driver assistance as well as camera-based assistance of the visually impaired, and present first findings concerning the development of such a transfer concept. The algorithms that result from the transfer concept have to be integrated into a camera-based assistive system. A sketch of such an assistive system is presented in [3] and [4] contains an overview of existing wearable assistive devices for the visually impaired.

In order to develop the transfer concept, we first gather the situations where visually impaired road users need support. We do this by combining procedures from Software Engineering, especially Requirement Engineering (RE), with methods from Social Sciences. As proposed by Sommerville in the chapter about RE [5], we conduct interviews, but whereas RE usually refers to a concrete system, we stay on a higher level of abstraction by discussing traffic situations for the visually impaired in general. For the interviews, we apply methods from qualitative Social Sciences, namely Expert Interviews [6] by means of Problem-Centered Interview methodology [7]. The evaluation of the interviews relies again on RE approaches. We use a slightly modified form of Scenarios as described in [5] to cluster the traffic situations mentioned in the interviews and from those Scenarios we derive Use Cases based on computer vision. Finally, we analyze which of the gathered Use Cases are addressed in ADAS and have to be further examined in the future in order to formulate the transfer concept.

The following chapter II describes the methodology we used. Afterwards, chapter III presents the current status of visually impaired road users by describing the Traffic Scenarios and derived computer vision Use Cases that result from the interviews. Before chapter V concludes the paper, chapter IV gives an overview of existing ADAS solutions concerning the identified Use Cases and presents an outline of our future work.

II. Methodology

After we explain why we conduct Expert Interviews and discuss the chosen method of Problem-Centered Interviews, the Interview Guideline is described. We conclude the chapter by giving an overview of the characteristics of the interviewees.

A. Expert Interviews

The aim of the Expert Interviews is to collect the main problems visually impaired persons face in traffic situations and furthermore to analyze Traffic Scenarios where a camera-based assistive system could provide support to visually impaired road users.

In this context, we define experts as persons who are regularly in contact through professional or volunteer work with a diverse group of visually impaired people concerning age, gender and degree of impairment. An own visual impairment of the experts is possible, but not a precondition. Rather than referring to the own experiences of the interviewees, Expert Interviews intend to access the ”contextual knowledge” [6] the interviewees have acquired concerning the living conditions of a certain group of people, in our case the visually impaired.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

As method for the Expert Interviews, we refer to Problem-Centered Interviews according to Witzel [7], a method that is settled between narrative and structured guideline interviews. During the interview, the sequence of the questions is handled flexible resulting in a dialogue between interviewee and interviewer.

Witzel names four required instruments: The short questionnaire's purpose at the beginning is to gather basic data about the interviewees, whereas the recording allows the later transcription of the interviews. The guideline serves as a useful reminder and guiding framework that assures the discussion of important issues and problems. Immediately following the interview, the interviewer writes a postscript containing nonverbal aspects, thematic peculiarities and spontaneous interpretative ideas for the evaluation.

Data analysis and interpretation were conducted by support of MAXQDA, a software program for qualitative data analysis, by which codes were developed in order to categorize specific Traffic Scenarios of visually impaired persons.

B. Interview Guideline

We create the guideline as a course for the interview under consideration of the four required instruments for Problem-Centered Interviews:

1) *Welcome and explanation of the aim of our research*
2) *Privacy policy* Includes the agreement of the interviewee to recording and transcription.
3) *Short questionnaire* Gather information about the interviewed person (age, gender, profession and if the work with the visually impaired is professional or voluntary) as well as about the visually impaired the interviewee is working with (age and gender distribution, kinds of visual impairment, affinity for technology).
4) *Traffic Scenarios* We first ask the interviewee to name the three biggest problems visually impaired people face in traffic situations and if there are differences in the problems for people of different age, gender and degree of impairment. By explaining problems, the interviewee usually mentions keywords for the discussion of further problems. If not, a prepared list with traffic situations (e.g. riding a bus, orientation at train stations, on the way from home to the bus station, finding an unknown address, differences between known and unknown ways) helps to give impulses to the interviewee. Towards the end of the interview, we ask about the differences between the visually impaired and the sighted when it comes to the preparation for a trip to an unknown address.
5) *Thanks and conclusion*
6) *Postscript*

C. Characteristics of the interviewees

In total, we interviewed four persons. All of them are male, their ages cover a range from over 40 to over 70. Three are blind, whereas the fourth person has no visual impairment. Two are active members of interest groups who work as volunteers and the others are employees of educational institutes. We use the following IDs to identify the interview partners in the course of this paper:

- **IP1:** male, blind, employee of educational institute
- **IP2:** male, sighted, employee of educational institute
- **IP3:** male, blind, volunteer of interest group
- **IP4:** male, blind, volunteer of interest group

Gender and degree of impairment of the visually impaired persons our interview partners work with are well distributed. Concerning age, the students of one of the educational institutes are generally not older than 19. But as the interest groups have more older members, due to demographic reasons and the fact that visual impairments are often age-related, we cover a wide range of ages.

The interviews were conducted via phone in German language. Therefore, the quotes in this paper are translations from German to English. All interviewees live in Germany, so that their statements refer to German traffic realities and may not be generally valid.

III. Current situation of visually impaired road users

This chapter presents the evaluation of the conducted Expert Interviews. First, we give some general Social Insights about the situation of visually impaired people. Afterwards, we define the Traffic Scenarios we extracted from the interviews by means of Software Engineering. For each Scenario, we derive those Use Cases that can be solved with computer vision.

A. Social Insights

We asked the interviewees about differences concerning gender, age or degree of impairment when it comes to affinity to technology and solving problems in traffic situations. Additionally, we wanted to know what the differences between the visually impaired and the sighted are when preparing for a trip to an unknown address and if many visually impaired people attempt such trips.

Only one interview partner, IP1, named a difference in gender. In his experience, girls and young women are more likely to attend voluntary events whose purpose is to provide additional advice and to pass on further knowledge, whereas boys and young men often show more confidence in their abilities and are less open to further advice.

According to the interviewees, age is less important to solve problems in traffic situations than life experience with visual impairment. The longer the people live with the impairment, the better they can usually handle those situations. But at the same time, an increased age also often causes further limitations, e.g. in hearing and motor skills.

Furthermore, the interviewees describe the community of the visually impaired as generally open to technological progress, be it out of interest or need: "When you have a limitation, you depend on technology and of course you use it." (IP4).

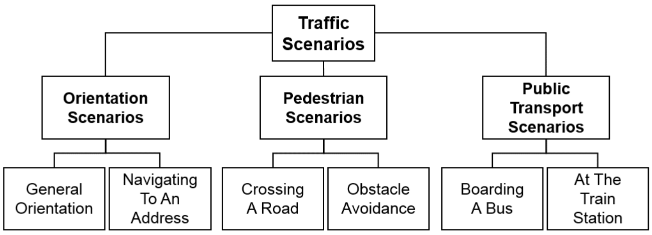


Fig. 1. Hierarchy of the identified Traffic Scenarios

When visually impaired people prepare for a trip to an unknown address, they essentially cover the same topics as the sighted, e.g. how to get there or what to consider, but the way they do it and the amount of needed information differ. The visually impaired also use online portals to check the timetables of public transport, but they have to rely on the accessibility of the websites. Even in case of accessibility, this can be a challenging task because most websites are optimized for visual perception. As maps are not accessible, the visually impaired have to either ask people who know the area or a sighted friend who can read the map for a description of the way and the area. These points make the preparation for a trip more time consuming for the visually impaired than for the sighted. One of the blind interviewees summarized it in the sentence ”I just simply need more precise information” (IP4), but of course it also depends on the people’s personality, no matter if they are sighted or visually impaired, how much and how detailed information they collect before going on a trip.

According to the impressions of the interviewees, a minority of the visually impaired attempt to travel to an unknown address on their own. It also depends on whether they have to travel the way only a few times or regularly. In case of regularity, they usually explore the way with an assistance person before they travel it alone. Providing the way has to be traveled only a few times, it depends on the complexity of the way and the amount of time that has to be spent if it is worth going alone. In the end, it is again a matter of personality how unknown ways are dealt with.

Although there are some differences between the sighted and the visually impaired, we can state in summary that ”the blind and visually impaired are as different individuals as you and your colleagues.” (IP4).

B. Traffic Scenarios

As can be seen in Fig. 1, we extracted a total of six Scenarios from the interviews. Those can be clustered into three categories: Orientation Scenarios (General Orientation, Navigating To An Address), Pedestrian Scenarios (Crossing A Road, Obstacle Avoidance), and Public Transport Scenarios (Boarding A Bus, At The Train Station).

We use the word Scenarios in the sense of Software Engineering defined by Sommerville in [5], but work on a higher level of abstraction. In contrary to [5], we do not refer to a concrete software system but discuss the challenges visually impaired people face as road users in general. Sommerville suggests to record Scenarios in tables with the keywords: Initial Assumption, Normal, What Can Go Wrong, Other Activities, and State Of Completion. As the last two are of no relevance in our case, we modify the table by deleting those keywords. In the added first line, Quotes, we introduce the Scenario by citing one of the interviewees to underline the importance of the described Scenarios. We use the line Normal to explain the current procedure to solve the Scenario and the line What Can Go Wrong to determine problems that can occur. In the added last line, Vision Use Cases, we record Vision Use Cases which can be solved by means of computer vision derived from the line What Can Go Wrong. This form of Scenario record, inspired by [5], gives us a clustered overview of the needs of visually impaired people in traffic situations. Furthermore, the gathered Use Cases are a starting point for the examination of ADAS algorithms that can be transferred to the domain of visually impaired road users.

TABLE I. General Orientation

Quote	”[It is] problematic in general to *keep an eye* on your destination. (...) You can simply get lost easily. (...) And this is a really burdening point for us.” (IP3)
Initial Assumption	A visually impaired person wants to know where they are and be aware of their direction and surroundings in order not to get lost.
Normal	Navigational smartphone apps, e. g. Blindsquare, help the person to keep track of their direction and also their surroundings as the app can announce crossings, shops, restaurants and such. On a smaller scale, Tactile Ground Guidance Systems (TGGS) help the person to find their way. If they get lost anyway, they can ask a passer-by or use the app BeMyEyes, which connects them via video chat to a seeing person who is willing to help.
What Can Go Wrong	The navigational app can only announce places in the database. If the data is not maintained or if the person is in a remote area, there might not be enough information for the person to create a mental map of their surroundings. Depending on the location of the person, GPS can be inaccurate. As TGGS are not directed, it is possible that they do not know, if they are walking it in the right direction. The person cannot find the TGGS even though it is there.
Vision Use Cases	TGGS Detection, Description of the surroundings, Traffic Sign Detection (e.g. to find pedestrian zones)

TABLE II. Navigating to an address

Quote	”[One] has to make sure to (...) find an entrance somewhere. Then, one has to ask someone, is this the right place or not?” (IP3)
Initial Assumption	A visually impaired person wants to walk to a certain address.
Normal	The person enters the address into a GPS based navigational smartphone app. The app leads them to the specified address.
What Can Go Wrong	Due to GPS accuracy, the navigational app cannot lead the person directly to the entrance of the building. If the building is unknown to them, they have to ask in order to get to the entrance and possibly find the right door bell.
Vision Use Cases	House Number Detection, Door Detection, OCR (Optical Character Recognition; for doorbell signs)

Two Scenarios concerning *Orientation* are extracted from the interviews: General Orientation, see Table I, focuses on self-localization and the awareness of the surrounding in order not to get lost, whereas the second Scenario, Navigating To An Address, see Table II, gives details on how to find a concrete house when trying to find an unknown address.

In addition to the discussed orientation issues, the *Pedestrian* Scenarios Crossing A Road, see Table III, and Obstacle Avoidance, see Table IV, are of high importance for the visually impaired.

When it comes to *Public Transport*, we identified the two Scenarios Boarding A Bus, see Table V, and At The Train

TABLE III. CROSSING A ROAD

Quote	”Crossing a road with high traffic frequency without any safeguarding is always a very big danger for a blind person.” (IP3)
Initial Assumption	A visually impaired person needs to cross a road.
Normal	With the help of TGGS or acoustic signals, the person finds a crosswalk or traffic light and safely crosses the road.
What Can Go Wrong	There is no TGGS in front of the crosswalk or the traffic light does not offer acoustic signals. The person cannot find the TGGS even though it is there. An intermediate platform may make them unsure on how to proceed. It can make them unsure, if they do not know the size of the road.
Vision Use Cases	Crosswalk Detection, Traffic Sign Detection (to detect the crosswalk sign), Traffic Light Detection, Traffic Light State Detection (red, green), TGGS Detection, Lane Detection (to extract information about the road’s size and course), Driving Vehicle Detection (to know if the road can be crossed)

TABLE IV. OBSTACLE AVOIDANCE

Quote	”They [obstacles] impede the walking flow, they interrupt you, you lose direction.” (IP4)
Initial Assumption	A visually impaired person is on the move as a pedestrian in traffic situations and has to take care not to collide with obstacles.
Normal	With the help of the white cane or a guide dog, obstacles are detected and avoided.
What Can Go Wrong	Whereas guide dogs are usually trained to detect ground as well as elevated obstacles, it is not possible to detect elevated obstacles with the white cane. The detection and avoidance of a construction site can be difficult. While moving around an obstacle, the person can lose orientation and/or drift away from the TGGS (see Table I).
Vision Use Cases	Obstacle Detection, Construction Site Detection, Traffic Sign Detection (to detect the construction site sign), TGGS Detection

Station, see Table VI. We neglected subways for two reasons: They are not present in the areas our interviewees are from and concerning Vision Use Cases, subways are a mixture of bus stops and train stations. Trams are not discussed because they are very similar to bus stops.

TABLE V. BOARDING A BUS

Quote	”The bus is also the most difficult means of transport, because it is so flexible.” (IP3)
Initial Assumption	A visually impaired person wants to board a bus.
Normal	The person waits at the bus stop on the entry field (marked with TGGS). When the bus arrives, they enter.
What Can Go Wrong	The person cannot find the bus stop They cannot find the entry field even though it is there. There is no entry field and the person has to rely on hearing to find the door. They do not know, if the arriving bus has the right number and direction and they might not want to ask every time. At a larger stop, where several buses stop at once, it is difficult to find the right bus.
Vision Use Cases	Traffic Sign Detection (to detect bus stop signs), TGGS Detection, Display Detection (to detect displays with important information), OCR (to read the text on the detected displays), Door Detection

TABLE VI. AT THE TRAIN STATION

Quote	”It [the train station] is easier to overlook and (...) at least at most train stations, there is some logic that you can understand.” (IP1)
Initial Assumption	A visually impaired person wants to travel by train.
Normal	A TGGS leads the person to the platforms. They find the right platform with the help of Braille indications under the handrails or they know the design of the station. Additionally, they can use the mobility service of the German railway company.
What Can Go Wrong	There is no TGGS leading to the platforms or the person cannot find the TGGS even though it is there. There are no Braille indications or they do not find the handrails. The mobility service is not available. Small train stops do not always offer announcements. They do not find the right track section which matches their seat reservation.
Vision Use Cases	TGGS Detection, Traffic Sign Detection (to detect platform numbers and platform section signs), Display Detection (to detect displays with important information), OCR (to read the text on the detected displays), Door Detection

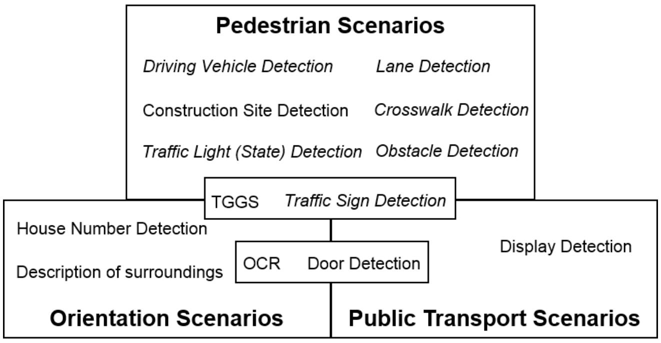


Fig. 2. Traffic Scenarios and their overlapping Use Cases. Italic Use Cases are of relevance in ADAS.

For our further examinations, we focus on the derived Use Cases. Therefore, Fig. 2 gives an overview of the Use Cases identified for the different kinds of Scenarios. We marked the Use Cases that are also of relevance in the field of ADAS in italic; those Use Cases will be discussed in the following chapter. From the figure, we can see that there are two Use Cases which are related to all kinds of Scenarios: Traffic Sign Detection and TGGS. Traffic Sign Detection is also addressed in ADAS and because of its relevance and versatility, it is the most important Use Case to transfer from ADAS. OCR and Door Detection are of interest in Orientation as well as Public Transport Scenarios, but are not discussed in ADAS. In summary, there are no ADAS Use Cases in Orientation or Public Transport Scenarios, but the already mentioned Traffic Sign Detection. In Pedestrian Scenarios, however, there is a large number and majority of Use Cases with relevance to ADAS, which makes it the most important group of Scenarios and Use Cases when it comes to transferring the according algorithms from ADAS to visually impaired road users.

IV. SOLUTIONS FROM ADAS AND FUTURE WORK

We first give a short overview of existing solutions for the ADAS Use Cases with relevance to visually impaired road users, as identified in the previous chapter. Afterwards, we present an outline of our future work.

A. Solutions from ADAS

In their chapter about related work, Yuan et al. [8] give an overview of existing methods for *Traffic Sign Detection and Recognition*. Traffic sign recognition usually consists of two blocks: detection and classification. In addition, tracking is used to increase the recognition rate. For detection, we have on the one hand methods based on color and shape (e.g. [9])

and on the other hand machine learning approaches, e.g. based on Support Vector Machines (SVM) [10]. Tracking detected signs with Kalman filters is applied with several intentions. For example, tracking can be used to include detection results from different frames [11] or to eliminate those results which cannot be found in successive frames [12]. As traffic sign classification is a typical object classification problem, the according algorithms are applied. Following feature extraction, SVM [10], Neural Networks [13], and other techniques are used. As those traffic signs who set out lane arrangements are often neglected by traffic sign recognition systems, [14] specifically addresses this problem.

A *Vehicle Detection* for urban areas under consideration of different angles in which the vehicle is recorded, is presented in [15]. The paper also contains an analysis of the start of the art for vision based vehicle detection sorted by certain characteristics such as on-road environment (highway, urban, night time), vehicle views (front, rear, side) or if it is covers partial occlusions and part based detection. Rubio et al. [16] for example present a vehicle detection for night time highways under consideration of front and rear views and without addressing partial occlusions or part based detection.

Vehicle detection can be seen as part of *Obstacle Detection*, as many obstacle detection developments in ADAS focus on the detection of specific objects. Besides vehicles, pedestrians and bicycles are detected (e.g. [17]). In those cases, a priori knowledge about the obstacles' texture, color, and shape is used in order to train models. On the contrary, Yang and Zheng [18] present a system that responds to every approaching object by exploiting motion patterns of a single dashboard camera. Other work that responds to all kinds of approaching objects can be for example found in [19], [20]. Those general methods are the ones that have to be considered for visually impaired pedestrians, because the kinds of obstacles they face are extremely distinctive.

Jung et al. [21] present a *Lane Detection* based on spatiotemporal images and summarize the general procedure of existing lane detection algorithms. Those algorithms consist of two phases: detecting the lane and fitting it to a parametric curve. In addition to edge detection followed by Hough transform, the main parts of the detection phase, some preprocessing steps, such as gray-level conversion, contrast or gradient enhancement, and noise reduction are executed. For the fitting phase, various interpolation methods such as B-spline, cubic or other polynomial curves are applied. For example, [22] introduces a lane detection method that first enhances the gradients and thus the lane marks. The lane is then detected by Canny edge detection and Hough transform, before a fitting based on a quadratic curve model is performed. To reduce occurring problems caused by changing illumination or missing parts of lane markings, motion vectors or optical flow are used (e.g. [23]).

In [24], a *Traffic Light (State) Detection* based on HSV color space, Maximally Stable Extremal Region (MSER), Histogram of Oriented Gradients (HOG) features, SVM, and Convolutional Neural Networks (CNN) is presented. Additionally, [24] describes the general structure of algorithms for traffic light (state) detection. First, candidates for traffic lights are identified by their color for which different color spaces can be used. The number of false candidates is then reduced by shape features, e.g. not round candidates are eliminated with the Hough transform [25]. Furthermore, structure information of the traffic light are extracted using global and local image feature descriptors like HOG [26] or Haar-like features [27]. The performance and robustness of traffic light detections is improved by using a combination of color, shape and structure features. For the recognition of the traffic light's state different classifiers such as SVM [26] or CNN [28] can be used.

For *Crosswalk Detection*, [29] proposes to extract crosswalk regions under different illuminations by means of MSER and to eliminate false candidates using Extended Random Sample Consensus (ERANSEC). The fact that crosswalks have a horizontal structure from driver perspective is used by [30] and [31]. Choi et al. [30] use a 1D mean filter and examine the difference image between original and filtered image, whereas Kummert and Hasselhoff [31] take advantage of the bipolarity and the straight lines of crosswalks by applying Fourier and Hough transform.

B. Future Work

As a foundation for our future research, it is essential to collect as many relevant Use Cases as possible. Therefore, we evaluate further Problem-Centered Interviews with visually impaired people where they focus on their own experiences. We already conducted and transcribed ten such interviews and their evaluation is ongoing. First results indicate that the results of this paper will be confirmed.

Since it is our objective to formulate a general concept to transfer camera-based traffic algorithms from ADAS to visually impaired pedestrians, we will focus on the overlap of Use Cases from those two fields, i.e. the ones marked in italic in Fig. 2 and possibly some others resulting from the interviews focusing on own experiences.

The before described solutions from ADAS are a starting point for an extensive literature review of relevant current ADAS algorithms in order to describe the state of the art. Those algorithms will be examined concerning the problems that occur when using them without any changes in the field of visually impaired road users. As we stated in [3], those problems occur mainly because of a priori assumptions such as known and stable camera position that are only valid when used in cars.

We will cluster the specified problems into different categories and in the next step develop a generalized concept to overcome the problems.

V. Conclusion

By combining methods from Social Sciences and Software Engineering, we conduct and evaluate Expert Interviews in order to gather those camera-based traffic Use Cases that can support visually impaired road users in their daily life.

Therefore, we extract different kinds of Scenarios from the interviews and derive the matching Vision Use Cases. The six identified Scenarios are clustered into three categories: Orientation, Pedestrian and Public Transport Scenarios. We extensively describe each Scenario based on the interviews and show that there is a large overlap in the Use Cases derived from the Scenarios between ADAS and the assistance of visually impaired people.

For the Use Cases belonging to both fields, we give a short overview of existing solutions from ADAS. In the future, we will extend this examination in order to identify and later overcome the problems that occur when using ADAS algorithms for visually impaired road users.

Besides giving us comprehensive insights about relevant Traffic Scenarios for the visually impaired, the interviews also encourage us to continue our research as one of the interviewees pointed out that "we currently have the rapid development of smartphones, and with that we are also experiencing more and more comfort. And in this context, such a development and research as yours is of utmost importance, so that one can achieve more safety in road traffic." (IP3).

References

[1] WHO Media centre. (2014, August) Visual impairment and blindness. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[2] H.-W. Wahl and V. Heyl, "Die psychosoziale Dimension von Sehverlust im Alter," *Forum für Psychotherapie, Psychiatrie, Psychosomatik und Beratung*, vol. 1, no. 45, pp. 21–44, 2015.

[3] J. Jakob and J. Tick, "Concept for transfer of driver assistance algorithms for blind and visually impaired people," in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Jan 2017, pp. 241–246.

[4] Á. Csapó, G. Wersényi, and M. Jeon, "A survey on hardware and software solutions for multimodal wearable assistive devices targeting the visually impaired," *Acta Polytechnica Hungarica*, vol. 13, no. 5, pp. 39–63, 2016.

[5] I. Sommerville, *Software Engineering*, 9th ed. Addison-Wesley, 2011.

[6] M. Meuser and U. Nagel, "Das Experteninterview - konzeptionelle Grundlagen und methodische Anlage," *Methoden der vergleichenden Politik- und Sozialwissenschaft*, pp. 465–479, 2009.

[7] A. Witzel, "Das problemzentrierte Interview," in *Qualitative Forschung in der Psychologie: Grundfragen, Verfahrensweisen, Anwendungsfelder.*, G. Jttemann, Ed. Beltz, 1985, pp. 227–255.

[8] Y. Yuan, Z. Xiong, and Q. Wang, "An incremental framework for video-based traffic sign detection, tracking, and recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1918–1929, July 2017.

[9] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition," in *Proceedings of the 16th Mini-EURO Conference and 10th Meeting of EWGT*, 2005, pp. 644–653.

[10] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," *Robotics and autonomous systems*, vol. 62, no. 1, pp. 16–24, 2014.

[11] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, June 2005, pp. 255–260.

[12] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognition*, vol. 43, no. 1, pp. 416–430, 2010.

[13] D. Cirean, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 1918–1921.

[14] Z. Fazekas and P. Gáspár, "Computerized recognition of traffic signs setting out lane arrangements," *Acta Polytechnica Hungarica*, vol. 12, no. 5, pp. 35–50, 2015.

[15] S. Sivaraman and M. M. Trivedi, "Vehicle detection by independent parts for urban driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1597–1608, Dec 2013.

[16] J. C. Rubio, J. Serrat, A. M. Lopez, and D. Ponsa, "Multiple-target tracking for intelligent headlights control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 594–605, June 2012.

[17] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito, and T. Mikami, "An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 942–953, Dec 2011.

[18] M. T. Yang and J. Y. Zheng, "On-road collision warning based on multiple foe segmentation using a dashboard camera," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 4974–4984, Nov 2015.

[19] W. Song, M. Fu, Y. Yang, M. Wang, X. Wang, and A. Kornhauser, "Real-time lane detection and forward collision warning system based on stereo vision," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 493–498.

[20] S. Santhanam, V. Balisavira, and V. K. Pandey, "Real-time obstacle detection by road plane segmentation," in *2013 IEEE 9th International Colloquium on Signal Processing and its Applications*, March 2013, pp. 151–154.

[21] S. Jung, J. Youn, and S. Sull, "Efficient lane detection based on spatiotemporal images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 289–295, Jan 2016.

[22] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1083–1094, Sept 2013.

[23] R. Sharma, G. Taubel, and J. S. Yang, "An optical flow and hough transform based approach to a lane departure warning system," in *11th IEEE International Conference on Control Automation (ICCA)*, June 2014, pp. 688–693.

[24] S. Saini, S. Nikhil, K. R. Konda, H. S. Bharadwaj, and N. Ganeshan, "An efficient vision-based traffic light detection and state recognition for autonomous vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 606–611.

[25] M. Omachi and S. Omachi, "Traffic light detection with color and edge information," in *2009 2nd IEEE International Conference on Computer Science and Information Technology*, Aug 2009, pp. 284–287.

[26] Q. Chen, Z. Shi, and Z. Zou, "Robust and real-time traffic light recognition based on hierarchical vision architecture," in *2014 7th International Congress on Image and Signal Processing*, Oct 2014, pp. 114–119.

[27] J. Gong, Y. Jiang, G. Xiong, C. Guan, G. Tao, and H. Chen, "The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles," in *2010 IEEE Intelligent Vehicles Symposium*, June 2010, pp. 431–435.

[28] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2014, pp. 2286–2291.

[29] Y. Zhai, G. Cui, Q. Gu, and L. Kong, "Crosswalk detection based on mser and eransac," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2770–2775.

[30] J. Choi, B. T. Ahn, and I. S. Kweon, "Crosswalk and traffic light detection via integral framework," in *The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, Jan 2013, pp. 309–312.

[31] A. Haselhoff and A. Kummert, "On visual crosswalk detection for driver assistance systems," in *2010 IEEE Intelligent Vehicles Symposium*, June 2010, pp. 883–888.

Examining Truthfulness of Informal Diagrams More Than a Year After Their Creation

Milan Jančár and Jaroslav Porubän
Department of Computers and Informatics
Technical University of Košice, Slovakia
Email: {milan.jancar, jaroslav.poruban}@tuke.sk

***Abstract*—Programmers document their software not only by formal, "official" means (UML diagrams, user and/or system documentation, wiki pages etc.) but also *ad-hoc* by drawing sketches, say *informal diagrams*, on a paper or a whiteboard. The question arises if these informal diagrams created at some time during development of a software system are still relevant and trustworthy after a time has passed. We believe that these diagrams are drawn mainly to capture important design decisions, which we assume to be durable. There is a problem regarding *evaluating the extent* to which a diagram is *true*. We proposed an approach for measuring truthfulness of such informal diagrams, and then, we conducted a mini-experiment on sample diagrams (aged 13–21 months) to measure their truthfulness. All five sample diagrams have truthfulness above the 40 % level, four of them even above the 70 % level. We do this to justify a further research of deeper inclusion of informal diagrams into the software development process.**

I. Introduction

Many developers spontaneously write or create notes, lists, tables, drawings, ER/class diagrams, etc. [1]; mainly "to understand, to design and to communicate" [2]. Many important design decisions are made on whiteboards [2]. There may be a resemblance to UML, but not strict [1]; a common denominator is *informality* and *spontaneity*. These sketches can be later changed, refined, even formalized (if necessary) – usually such sketches also have a longer lifespan and are more likely to be archived [3]. An interesting fact is what medium developers use for sketching: almost two thirds are on some analog medium (mainly paper and whiteboards), whereas modern means such as *interactive* whiteboards, tablets and smartphones are almost never used [3].

The notion of an *informal diagram* or, shortly, a *diagram* is used in this paper to refer to UML-like diagrams, or any informal diagrams without a priori defined notation, or any sketch, drawing, or even just a note containing some important piece of information relevant to a related software system.

We collected 30 informal diagrams capturing internals of a commercially developed system we participate on. The added value of our collection lies in the fact that they are *quite old* – their origin is in range from *two years ago* to *a year ago* (October 2015 – October 2016).

A. Problem – are diagrams still trustworthy?

We are interested in the following research question: *Are facts captured in an informal diagram to significant extent true even more than a year after the diagram's creation?*

We think they might be. The reasoning behind this is that people capture mainly important design decisions and these tend to be more durable.

B. Our approach – measuring "truthfulness"

Our approach is to quantify a so called *truthfulness* of sample diagrams. *What* do we mean by this term is explained here. *How* to quantify this measure, i.e. our approach, is described in the Section III.

Let us have a software system and a set of facts which are true about it at some point in time (a set S, see Fig. 1). S is the ultimate *truth*. We also have informal diagrams $(d_1, d_2, \ldots)$ which convey information $(D_1, D_2, \ldots)$ about the system. These diagrams were created over time and captured the truth about the system in a particular time of their creation.

As the system evolves and, therefore, the set of facts describing the system changes, the existing diagrams might not reflect the current state properly – a diagram may convey information which is still true, but also it may hold an outdated piece of information.

Fig. 1 depicts this situation on three example diagrams. The set of true facts S about the system is, obviously, incomparatively larger than any set of facts D_i conveyed by an arbitrary diagram. There may be diagrams that are still completely true (D_3) or rather true (D_1), some are rather not (D_2). It is possible that a diagram captures facts captured also by another diagram (D_1 and D_3).

When we mention *truthfulness of a diagram* ($t(d_i)$), we mean an *extent* (hence no binary logic) to which the diagram is true, and we express it as a ratio of an amount of *true* facts about the system conveyed by the diagram to an amount of *all* facts about the system conveyed by that diagram; symbolically:

$$t(d_i) = \frac{|D_i \cap S|}{|D_i|}$$

Note that we are picking pieces of information from diagrams (elements of D_i) and checking if they are true ($\in S$) and not reversely: checking if true information are present in a diagram, which would be much more complicated.

C. Contribution

The contribution of this paper is twofold:

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

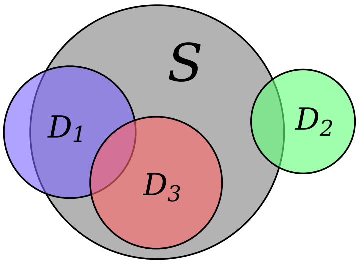


Fig. 1: An Euler diagram depicting a coverage of a system by informal diagrams

1) addressing the question of trustworthiness of documentation after a time has passed by *evaluating the truthfulness* of old informal diagrams documenting a software system;
2) *presenting an approach* for quantifying the extent to which some unstructured artifact (an informal diagram in our case) is true about its related software system.

We realize that it may be hard to draw some generally valuable conclusions from our mini-experiment (1); therefore, the emphasis is put on explaining our approach (2).

II. Background

As mentioned in the Introduction, we collected 30 informal diagrams capturing internals of a commercially developed system we participate on and they are aged 1–2 years. They are private notes taken by one of the authors of the paper.

A. Informal diagrams

It is important to note that diagrams were *not* intended for such a research. Their studying is purely incidental. Knowing about the consecutive research beforehand might have invalidated obtained results because the sketches might have been (subconsciously) adjusted or a greater effort might have been put into creating them (cf. Hawthorne effect).

The diagrams contain informal instances of the following types [4]: an entity–relationship (class) diagram (30 %), an algorithm (23 %), a user interface sketch (17 %), a state transition diagram (13 %), a table/matrix (13 %), a question/answer list (13 %), modules structure and their dependencies (7 %), and generic notes (60 %).

More about *time-framing* and the *nature* of these diagrams can be found in our previous work [4].

B. Software System

The software system documented by the mentioned diagrams has the following nature. The system is a company internal system for managing their customers (people applying for a job). It is a web application consisting of a server side created in Grails (a Groovy-based framework for the Java Virtual Machine) and a client side created in AngularJS (a JavaScript framework). Source files have ca. 150 KLOC[1] and following programming languages are used[2]: Groovy (48%), JavaScript (28%), HTML (14%), Java (7%) and some others.

III. Measuring truthfulness of informal diagrams

As mentioned in the introduction, we have a software system, the *source code* of which we consider the ultimate truth. On the other hand, we have possibly outdated *informal diagrams* about this system. We put these two artifacts against each other and there are at least three problems about *comparing* these two:

1) The source code is incomparably larger than diagrams; therefore, we cannot measure their "equality", only "coverage" of the code by diagrams (as explained in the Introduction, we consider this problem solved; see Fig. 1).
2) Diagrams are *informal*, which means there is no a priori defined notation, nor a structure, hence no automated way of extracting information from it.
3) Diagrams are *heterogeneous*, each one is somewhat unique. Therefore, it is even harder to find a common way which would fit to all informal diagrams.

For a comparison, let us consider we have two UML diagrams as opposed to our code–diagram situation. They are roughly about the same size (ad 1), UML diagrams have some predefined structure (ad 2), and both are of the same type (ad 3). There are approaches for comparing such artifacts, e.g. by transforming them to trees (which is straightforward) and then comparing these trees by top-down or bottom-up approach [5].

If we had diagrams of the same (or similar) type, say a class diagram, we could prepare some evaluating *scheme*, e.g., our scheme would notice the following *aspects*: names of classes, attributes, relationships, etc. Then we would decide on how to regard these aspects individually.

However, that is not the case. Thus, we must abstract our hypothetical scheme to a level where no concrete aspects exist; we can just state that a diagram contains *a* fact – we are, in general, not able to say anything more concrete about that fact.

This reasoning leads to a principle of our approach: deconstructing an arbitrary given informal diagram to a *set of facts*, ideally *atomic* assertions that can easily be evaluated as true (1) or false (0). Overall truthfulness of a diagram is then calculated from these values (1 or 0) of these particular facts. How exactly this is done is elaborated in the following subsection. Note that we will use the notion of a *fact* in the above-mentioned sense.

A. Weighting and decomposing facts

We have already suggested in the Introduction what the *truthfulness* of a diagram should express – a ratio of number of *true* facts (included in a given diagram) to number of *all* facts (in that diagram).

[1] Kilo Lines of Code (no blank or comment lines), based on the output of `cloc` tool applied to files stored in a Git repository (`git ls-files`)

[2] As reported by *GitLab* in Graphs–Languages section

Fact	*Weight*	*True?*	*Lower*	*Upper*
There is a class named Applicant	2	1	2	2
… CommLog	2	1	2	2
… CommState	2	1	2	2
… ComLogEntry	2	1	2	2
… LeadImportLogEntry	2	1	2	2
… AssignLogEntry	2	1	2	2
… ContactLogEntry	2	0	0	0
There is a 1..1 relationship between Applicant and CommLog	2	1	2	2
There is a 1..n relationship between CommLog and ComLogEntry	2	1	2	2
ContactLogEntry extends ComLogEntry class	2	0	0	0
AssignLogEntry extends ComLogEntry class	2	1	2	2
LeadImportLogEntry extends ComLogEntry class	2	1	2	2
CommLog has a logEntries property	1	1	1	1
CommLog has a state property	0.5	1	0.5	0.5
… And its name is “state“	0.5	0	0	0
… And it refers to one CommState object	1	1	1	1
CommLog has “autostamp“ properties	1	1	1	1
“Autostamp“ properties are: dateCreated, lastUpdated, createdBy, editedBy, and NO other	5	1	5	5
ComLogEntry has a note property	1	1	1	1
ComLogEntry has the “autostamp“ properties	1	1	1	1
CommState has a name property ♦	0.5	1	0.5	0.5
… And its name is “name“	0.5	0	0	0
… And it may have a value: Unassigned	1	0	0	0
… … Assigned	1	0	0	0
… … Contacted	1	1	1	1
… … and no other ■	1	0	0	0
LeadImportLogEntry has source, sourceDomain, preferredPosition, jobOrder, and importId properties	5	1	5	5
LeadImportLogEntry has an originalDate property	1	0	0	0
LeadImportLogEntry is marked as #1	1	• ?	0	1
AssignLogEntry is marked as #2	1	?	0	1
ContactLogEntry is marked as #3	1	?	0	1
AssignLogEntry is not dedicated only for “leads“	1	0	0	0
ContactLogEntry is not dedicated only for “leads“	1	0	0	0
$\sum$	51		**72.5 %**	**78.4 %**

TABLE I: Example of the reviewed facts in the diagram # 20 (created: jan/feb 2016)

However, to make this ratio representative, we must take into account that a given fact (from a set of facts for a given diagram) is not necessarily as “big” as another one (from the same set). To accommodate this aspect, we assign each fact a *weight*. There are two reasons for weighting.

First, objectively, one fact can convey more information than another one. To demonstrate this, let us consider the following example facts:

- (A) “There is a class X.” – deduced from a drawn rectangle with a “X” at the top of it;
- (B) “The class Y extends the abstract class Z.” – deduced from two appropriately labeled rectangles connected by an arrow.

The fact B can be decomposed to the following assertions:

- (B_1) “There is a class Y.”
- (B_2) “There is a class Z.”
- (B_3) “The class Z is abstract.”
- (B_4) “The class Y extends the class Z.”

Apparently, the assertions A, B_1, and B_2 have the equal information value (because of the same sentence structure). Intuitively, B_3 and B_4 also carry around the same information value. Thus, B's weight should be four times the weight of A.

The second reason for weighting is rather subjective – we need to regard the *importance* of a given fact. Unfortunately, we do not know how to objectively rate the importance. Thus, the fact that conveys two times the information value than another fact can end up with the equal weight if it is of a half the (perceived) importance.

We conclude these considerations about weighting with a formula for calculating *truthfulness* of a diagram. For a given diagram d_i, let us have a set D_i of facts f_{ij} conveyed by the diagram. Each fact is then given a weight $w(f_{ij})$ and checked against the source code of the related software system if it is true ($t(f_{ij}) = 1$) or false ($t(f_{ij}) = 0$). The truthfulness of the diagram is then calculated intuitively as follows:

$$t(d_i) = \frac{\sum_{f_{ij} \in D_i} w(f_{ij}) \cdot t(f_{ij})}{\sum_{f_{ij} \in D_i} w(f_{ij})}$$

The value ranges from 0 to 1 inclusively (thus, it can be expressed as a percentage), and it is just a refinement of how $t(d_i)$ was defined in the Introduction section. It is simply a ratio of sum of weights of true facts to sum of all weights (for a given diagram).

B. Atomicity of facts

It is advantageous to decompose facts to as terse (one might say "atomic") assertions as possible for two reasons: (1) it is easier to weight them, and (2) it is easier to evaluate them as true or false (since complex facts might be partially true). For example, stating that a fact is 50 % true is de facto the same as decomposing it to two facts of a half the weight and considering one of them true and another false.

As seen in the Table I, we have not always stuck to the rule when we were able to evaluate a complex fact as a whole. However, if we extracted a complex fact from a diagram and were not able to evaluate it as a whole, then we split it to more terse facts; of course, the sum of their weights must have been equal to the weight of the original fact.

On the other hand, there were cases (see a row in Table I marked ⧫) when we split an already terse fact of a type

- "There is an entity X."

into two "subfacts", each having a half the weight:

- "There is an entity ◇." – still true;
- "◇'s name is X." – now false.

It means *the entity* (now identified by an abstract symbol ◇) still exists, but it has been *renamed*. We now distinguish between the fact of *existence* and fact of *naming*. It is useful when there are other facts about that entity such as "X has a property P." – we do not want to evaluate them as false just because the symbol X cannot be found in the code any more, so we read these facts like "◇ has a property P.", which might be still a true fact. For the sake of simplicity, in the Table I, we refer to such entities still by their old name, but, as explained, we mean *the entity* previously connected with that name (◇), *not the name* itself.

Although we try to keep a process of evaluating the truthfulness of facts in a diagram as "mechanic" as possible, it requires a little heuristic to distinguish between an entity having been renamed from X to Y and the fact that the entity X has been deleted and a new independent entity Y now exists. We consider an entity X being renamed to Y basically in the two cases: (1) there are other facts describing the X that now fit the Y, and (2) we see it intuitively when the new name does not differ too much.

C. Interpreting omission

One thing that needs a consideration is how to interpret an *omission* of something in a diagram. Some examples follow:

1) We have a (pseudo-class) diagram containing some rectangles representing classes, but *no* methods or fields are listed in *none* of the rectangles.
2) We have a diagram containing some rectangles representing classes – some of them list their fields (or methods), while *some others do not.* (See Fig. 3.)
3) We have a diagram that lists values of an enum types. (See Fig. 3.) – The code contains the listed values, but it also contains some extra values, *omitted* in the diagram.

Does the omission of a respective entity in a diagram mean its *non-existence* in the code? Or does it not mean anything and the omission is due to the entity being *not important*?

Ad (1): It is reasonable to say that omission does not mean anything in this case. We can assume that classes contain some methods and they were just not listed. If we claimed the opposite (no methods in a diagram mean no methods in the code), then we could argue *ad absurdum* that, e.g., not mentioning access restrictions in *that* diagram means there are none restrictions in the implemented system.

Ad (2): If we take into account that we deal with *informal* diagrams with no predefined structure, then we cannot assume *anything* about the type of a given diagram (a similarity to a class diagram at the first glance is not enough); therefore, we cannot really require a strong consistency across that diagram. Thus, again, omission should not be perceived as non-existence.

In this two cases we do not consider omission to be erroneous. Omitted facts were neglected or considered not important. The truthfulness of a diagram is left intact and we are consistent with our claims from the Introduction section that, put simply, we check if things from a diagram are present in the code and not vice versa.

Ad (3): This is an example of a situation when the omission of an element in a diagram probably *should* imply its non-existence, and thus, the diagram is erroneous. This case includes situations when there is a "common sense" to list *all* elements, such as lists of possible values of an enum type, transitions in a state transition diagram, etc.

This distinction is important because a set of facts of the mentioned diagram (3) will not include only facts like "The enum type X includes the x value," but also one latent fact – "There are no other values in the enum type X." (see ■ in Table I). This additional fact will be false, thus the truthfulness of a diagram will be lowered, as we would intuitively expect.

D. Lower and upper bounds of truthfulness

For a given diagram, we calculated its truthfulness not as a single value (from 0 % to 100 %) but as a *range*, e.g. 73 %–78 %. The reason is simple: for some facts derived from the diagram, we were not able to determine if they are true or false, i.e. $t(f_{ij}) = ?$ (see the Table I, a • mark). So we used the formula for truthfulness (cf. Section III-A) two times: the first time we considered all undecided facts false, thus getting a lower, more pessimistic value (a lower bound, T_{min}), and the second time we considered them all true, thus getting a higher, more optimistic value (an upper bound, T_{max}).

E. Interdependence of facts

Consider two facts: "X contains just y and z," and "X contains just two elements." The second fact can be inferred from the first. Truthfulness of such a facts is mutually bound. Including redundant facts to the set of facts D_i of a particular diagram may affect the calculated truthfulness if, for example, we include redundant facts derived from true facts but not from false facts (we would de facto manipulate the weight of true facts). It is reasonable to say that a fact is redundant if we can omit it from the set and still are able to reconstruct the diagram to its full original form, based only on the facts remaining in the set.

Diagram #	*Age**	T_{min}	T_{max}
7	21.0	43.6%	46.2%
20	19.5	72.5%	78.4%
14	19.0	80.7%	84.2%
29	15.0	88.0%	100.0%
36	13.5	82.9%	87.4%

* in months (rounded to a half a month);
reference date: 2017/09/09

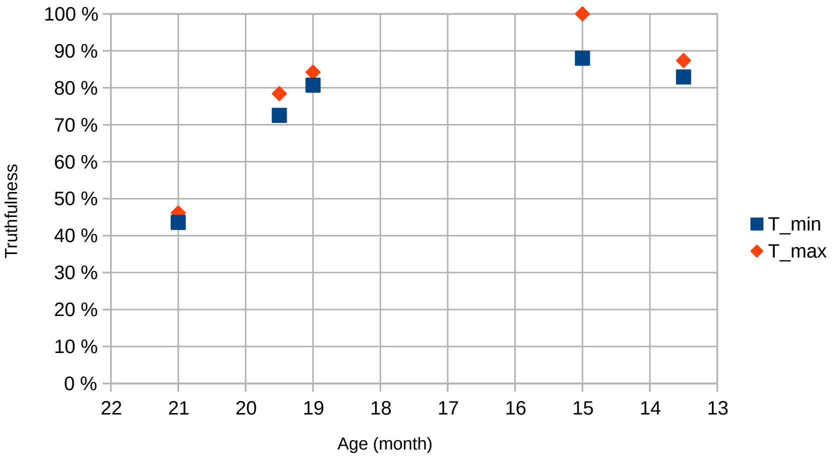


Fig. 2: Calculated *truthfulness* of sample diagrams (from left to right): # 7, 20, 14, 29, 36

F. A few additional notes

- Abbreviations are massively used in diagrams; their usage is not considered erroneous, e.g. if a diagram contains a reference to "CommSvc" and the actual name in the code is "CommunicationService", it is acceptable.
- A diagram might contain a legend (an explanation of a notation and abbreviations) – we treat this as meta-information – it is information at another level, obviously truthful, there is no use in evaluating it.
- Diagrams may contain notes not directly describing the system. They may include various considerations, universally valid claims – these were not evaluated; or even questions – it makes no sense to evaluate (as true or false) those.

IV. Results

By the method elaborated in the previous section, we evaluated five sample diagrams. Results can be seen in the table and the graph in Fig. 2. All five sample diagrams have truthfulness above the 40 % level, four of them even above the 70 % level. For the diagram # 20, complete data are provided: a photo of the diagram (see Fig. 3) and the table capturing the evaluating process (see Table I). (There was no room to include data for all the sample diagrams.)

V. Discussion

The overall average truthfulness of the sample diagrams is 76.4 % – ca. three quarters. Also a slight tendency of older diagrams to be less truthful can be seen in the graph in Fig. 2. It is hard to clearly answer our research question; it requires a further research.

A. Threats to validity

Major threats are the following:

- The collection of five chosen diagrams is not very representative.
- The sampling method (choosing 5 diagrams from 30 available) was based on our convenience.
- Even worse, the studied diagrams are of just one person and are related to one software system.

Therefore, we do not dare to draw any strong conclusions.

Minor threats include: subjective weighting of facts, assuming that we are able to interpret a notation and abbreviations used in diagrams as originally intended.

VI. Related Work

Regarding our studying of informal diagrams: Mangano et al. [1] and also Cherubini et al. [2] studied the interaction of software designers (developers) with sketches at the whiteboard; an exploratory study of sketches and diagrams was presented by Baltes & Diehl [3].

The paper [6] describes "formalization of UML class diagram." It is only marginally connected to our work, since we are not constrained to the UML kind of diagrams and also, we have not required a strict formalism. However, the paper provided some insights on features that should be considered during a diagram analysis/decomposition.

In our approach, we were basically comparing two artifacts: a diagram and a source code. Kelter et al. [5] dealt with comparing two UML models by transforming them into trees first.

VII. Conclusion

Informal diagrams (or "sketches") definitely have their place in the software development. This claim is supported by the existing research in this area, mentioned in the Introduction and the Section VI.

One of interesting things to study about these informal diagrams, which are created spontaneously during the development of a software, is how trustworthy they are after a time passes. We have a collection of diagrams aged more than a year. This collection might be considered valuable because in

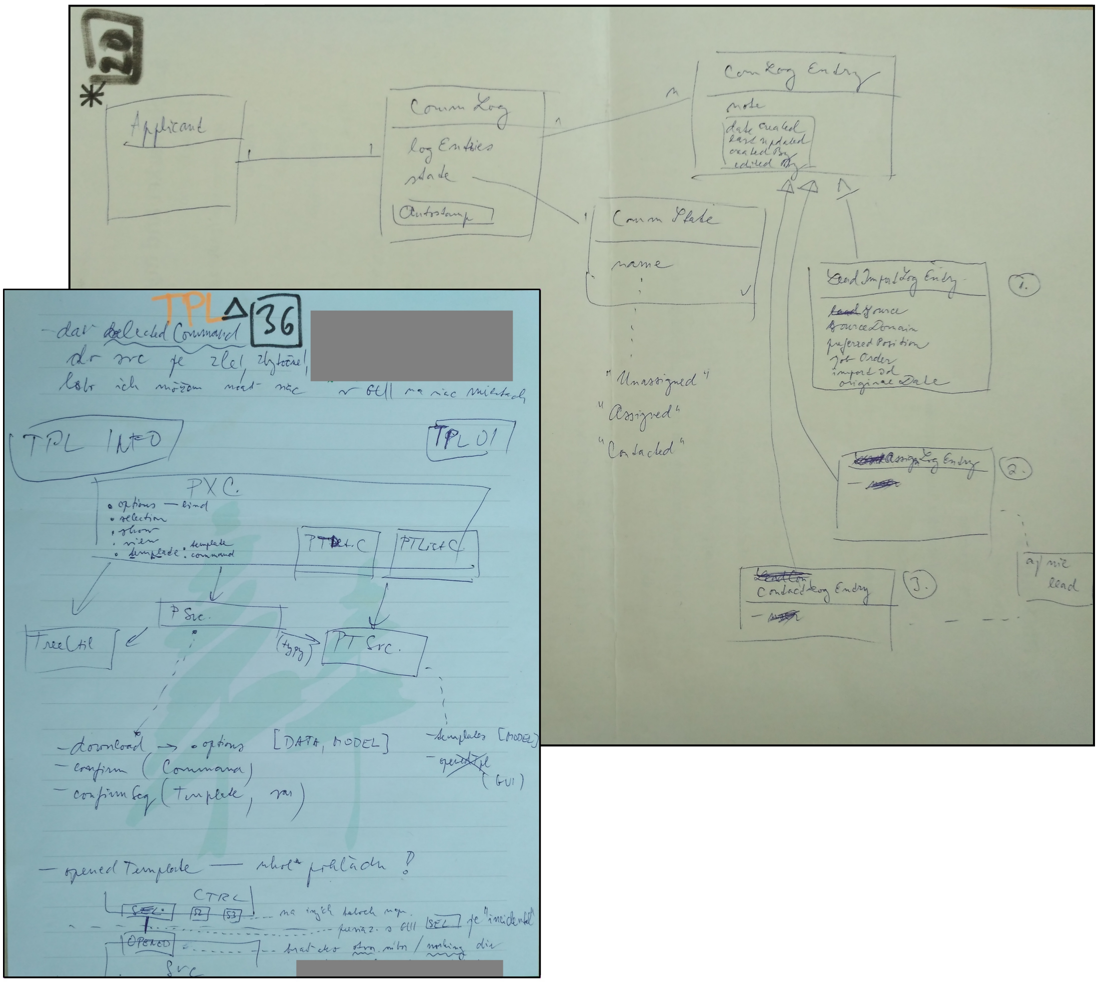


Fig. 3: Sample diagrams # 36 and 20

the time of creation of these diagrams, there was *no intent* of their further studying, hence no bias; diagrams are as natural as possible. Yet, evaluating truthfulness of informal artifacts, which have no prescribed structure or semantics, rendered *hard*. We have presented an approach for such an evaluation. The presented experience is probably the most valuable contribution delivered by this paper.

Acknowledgment

This work was supported by the project KEGA No. 047TUKE-4/2016: "Integrating software processes into the teaching of programming" and by the project FEI-2015-23: "Pattern based domain-specific language development".

References

[1] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek, "How software designers interact with sketches at the whiteboard," *IEEE Transactions on Software Engineering*, vol. 41, no. 2, pp. 135–156, Feb 2015.

[2] M. Cherubini, G. Venolia, R. DeLine, and A. J. Ko, "Let's go to the whiteboard: how and why software developers use drawings," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 557–566.

[3] S. Baltes and S. Diehl, "Sketches and diagrams in practice," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 2014, pp. 530–541.

[4] M. Jančár and J. Porubän, "Towards Employing Informal Sketches and Diagrams in Software Development," in *6th Symposium on Languages, Applications and Technologies (SLATE 2017)*, ser. OpenAccess Series in Informatics (OASIcs), vol. 56. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, pp. 4:1–4:10.

[5] U. Kelter, J. Wehren, and J. Niere, "A generic difference algorithm for uml models." *Software Engineering*, vol. 64, no. 105-116, pp. 4–9, 2005.

[6] Z. Xu, Y. Ni, W. He, L. Lin, and Q. Yan, "Automatic extraction of owl ontologies from uml class diagrams: a semantics-preserving approach," *World Wide Web*, vol. 15, no. 5, pp. 517–545, 2012.

Energy Load Forecast Using S2S Deep Neural Networks with k-Shape Clustering

Tomáš Jarábek
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
tomas.jarabek@stuba.sk

Peter Laurinec
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
peter.laurinec@stuba.sk

Mária Lucká
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
maria.lucka@stuba.sk

***Abstract*—Ensuring sustainability demands more precise energy management to minimize energy wastage. With the deployment of smart grids that provide a huge amount of data, new methods of machine learning come to light to ensure more precise predictions. With the huge amount of data, deep learning methods which require a huge amount of data, are starting to show promise with increasing accuracy. In this paper, we present a methodology for predicting time series which uses Deep Neural Networks, specifically Long Short-Term Memory (LSTM) algorithm with a Sequence to Sequence (S2S) architecture. We improve the prediction accuracy, when predicting the aggregated load of the network, by using the k-Shape clustering algorithm to create groups of consumers that are easier to predict. The methods described in this paper were tested on real world data of electricity consumption of Slovak enterprises.**

***Index Terms*—deep learning, LSTM, S2S, k-Skape Clustering, Electricity Consumption Forecast**

I. Introduction

Accurate forecast methods of electricity consumption are widely studied, because of economic, environmental and technical reasons. Because of technological progress, smart grids consisting of many smart meters are being introduced into everyday life, enabling the collection and usage of a large amount of data about consumer electricity demand, helping to improve the demand forecast. This helps to reduce the need for excessive electricity production and thus the burden to the environment. Electricity consumers generally have a stochastic behavior, making individual consumer demand hard to predict. To counter this problem the consumption data is aggregated. For that reason, cluster analysis is used [1] to classify consumers to more predictable groups. Cluster analysis in the energy domain can be also used in other use cases, such as creation of consumer profiles, detection of anomalies, emergency analysis, dynamic pricing etc. An individual consumer electricity demand can be represented by a time series of real valued measurements. The application of clustering to time series of electricity consumption can be used for dimension reduction to reduce memory requirements and computational complexity, for removing noise in the data and to emphasize the important characteristics of the data.

The load forecast at both, aggregate and building level received the increased attention of researchers. In literature two main techniques can be found for forecasting energy load demand: 1) Models based on physics principles and 2) Statistical and machine learning based models. This work is focused on the second technique. In [2] the authors used a Recurrent Neural Network (RNN) with Long Short-term memory cells and compared it to one with a Sequence to sequence (S2S) architecture in predicting building level energy demand. They showed that the network with the S2S architecture outperformed the standard LSTM one.

In this paper, we show the effectiveness of using the k-Shape clustering algorithm in conjunction with the S2S architecture to predict aggregate level energy demand without using other data except the energy load measurements. This generalizes the outcome to any collection of time series even outside the energy domain.

II. Methods

This section introduces the dataset used for the testing, and presents the approach for addressing the problem. It also introduces the clustering algorithm used in the experiment.

A. Dataset details

The methods and experiments introduced in this paper have been applied to a dataset, comprising of data from smart meters about Slovak electricity consumption. This data was collected within the project International Centre of Excellence for Research of Intelligent and Secure Information-Communication Technologies and Systems from Slovak enterprises. The data was recorded from the 1st of July 2013 to the 16th of February 2015 with a resolution of 15 min. The consumption of 11281 enterprises was aggregated into 1152 time series according to the postal code of the enterprise. In this paper, these aggregated time series will represent the consumers of the network. This idea is something akin to the local consumption of a small area. The dataset was divided into 3 parts with the approximate ratio of 70-15-15: The training set from 01-07-2013 to 19-08-2014, the validation set from 20-08-2014 to 16-11-2014 and the testing set from 17-11-2014 to 16-02-2015. Only the data of the load demand was used for the prediction without other cyclic features to better isolate the experiment.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

B. Problem statement

The smart meter records consist of a set of M consumers $\boldsymbol{x_1}$, $\boldsymbol{x_2}$, ..., $\boldsymbol{x_M}$. The load demand of a consumer i can be defined as $\boldsymbol{x_i} = \{x_{i,1}, x_{i,2}, ..., x_{i,T}\}, \forall i \in \{1,2,...,M\}$ where T is the total number of time periods and M is the total number of consumers. Our aim is to predict the total aggregated load of the network. There are three approaches for the load prediction of network level demand that have been reported in literature. First, the aggregated approach [3] which aggregates the demand of all consumers in the network and takes it as the input for the forecasting. Secondly, there is the approach to predict each individual consumer demand and then to sum up these individual predictions to get the network level demand forecast [3]. The third approach is by clustering individual consumers into specific clusters to generate predictions for these individual clusters. The total demand is then predicted by taking the sum of these predictions. This approach is similar to the approach in [4]. In our experiment, we compare the first and the third approach when using a deep learning recurrent neural network.

C. k-Shape clustering

Many clustering algorithms have been used for the load prediction problem. Determining the best algorithm to be used depends on the nature of the problem. In this paper, we use the *k*-Shape [5] algorithm for clustering time series, with a distance measure that is invariant to scaling and shifting. It is a centroid based clustering algorithm that can preserve the shapes of time-series sequences.

First, we discuss the distance measure that is based on the cross-correlation measure. We introduce a method to compute the centroids of time series clusters which is based on this measure. Finally, we describe the *k*-Shape clustering algorithm that efficiently generates clusters of time series.

1) Shape-Based Distance: The first component of the algorithm is the Shape-Based distance that is based on the cross-correlation measure. Cross-correlation measure is a statistical measure with which the similarity of two sequences can be determined. Consider two sequences $\boldsymbol{x} = \{x_1, x_2, ..., x_m\}$ and $\boldsymbol{y} = \{y_1, y_2, ..., y_m\}$. The degree of similarity of these sequences can be characterized as $CC_w(\boldsymbol{x,y}) = R_{w-m}(\boldsymbol{x,y})$, for $w \in \{1,2,...,2m-1\}$, where $CC_w(\boldsymbol{x,y})$ is the cross-correlation sequence of length $2m$-1, and $R_{w-m}(\boldsymbol{x,y})$ is computed, as:

$$R_k(\boldsymbol{x,y}) = \begin{cases} \sum_{i=1}^{m-k} x_{i+k} y_i, & k \geq 0 \\ R_{-k}(\boldsymbol{y,x}), & k < 0 \end{cases} \tag{1}$$

For scaling invariance, the algorithm uses Z-score normalization, where it transforms each sequence $\boldsymbol{x}$ into $\boldsymbol{x'} = \frac{x-\mu}{\sigma}$ so that its mean μ is zero and standard deviation σ is one. Depending on the domain or the application, different normalizations for $CC_w(\boldsymbol{x,y})$ might be used. The algorithm uses the coefficient normalization which is defined as follows:

$$NCC_c(\boldsymbol{x,y}) = \frac{CC_w(\boldsymbol{x,y})}{\sqrt{R_0(\boldsymbol{x,x}) R_0(\boldsymbol{y,y})}} \tag{2}$$

This also addresses shift-invariance by generating values between [-1,1]. The goal is then to find the position w where $NCC_c(\boldsymbol{x,y})$ is maximized. This way we get the following Shape Based Distance (SBD):

$$SBD(\boldsymbol{x,y}) = 1 - \max_{w} \frac{CC_w(\boldsymbol{x,y})}{\sqrt{R_0(\boldsymbol{x,x}) R_0(\boldsymbol{y,y})}} \tag{3}$$

that takes values between 0 to 2 with 0 indicating perfect similarity for the given time series sequences.

2) Time Series Shape Extraction: The second part of *k*-Shape clustering is the extraction of a centroid, for a set of time series sequences, considering the SBD distance measure. Archiving accurate shape extraction of the centroid changes this to an optimization problem where the objective function is to find the maximizer $\boldsymbol{\mu_k}^*$. It can be found by solving the following equation:

$$\boldsymbol{\mu_k}^* = \underset{\boldsymbol{\mu_k}}{\operatorname{argmin}} \sum_{\boldsymbol{x_i} \in P_k} (NCC_c(\boldsymbol{x_i}, \boldsymbol{\mu_k}))^2 \tag{4}$$

where P_k is the k^{th} partition and $\boldsymbol{\mu_k}$ is the initial centroid for the k^{th} partition. The equation uses the previously computed centroid as reference to align all sequences towards this reference sequence for each iteration of the algorithm. The SBD identifies for each sequence the optimal drift towards the centroid for every $\boldsymbol{x_i} \in P_k$.

3) Shape-based Time Series Clustering: The *k*-Shape algorithm for time series clustering relies on the SBD and the shape extraction method to produce clusters of time series. *k*-Shape expects as input the set of time series X and the number of clusters k that we want to produce. First the algorithm randomly assigns the time series in X to clusters. Then it computes the centroid for each cluster using the shape extraction method. In the next step, it refines the memberships of the clusters by using the SBD distance measure. The algorithm repeats until it converges or until it reaches the maximum number of iterations (usually a small number, such as 100). The output of the algorithm is the assignment of sequences to clusters and their corresponding centroids which show the most representative shape of each cluster.

III. Neural Network Architecture

In this section, we introduce our deep learning architecture that was used in the experiment. In the first part of this

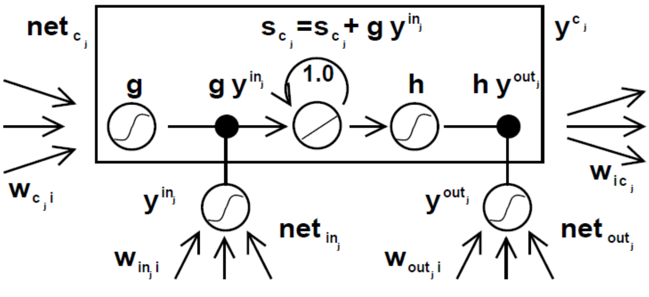


Fig. 1: Architecture of a LSTM memory cell and its gate units [4].

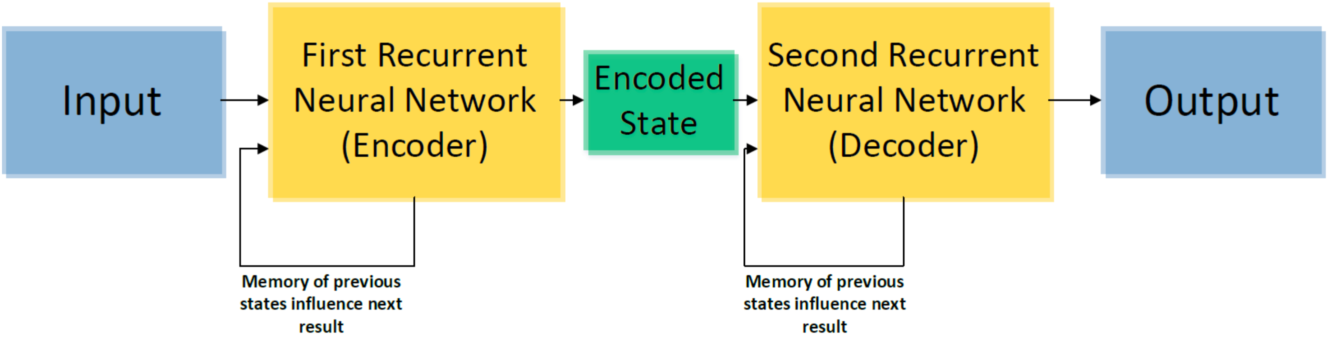

Input
First Recurrent Neural Network (Encoder)
Encoded State
Second Recurrent Neural Network (Decoder)
Output
Memory of previous states influence next result
Memory of previous states influence next result

Fig. 2: Diagram of the S2S based architecture.

section the Long Short-Term Memory (LSTM) algorithm for training recurrent neural networks (RNN) is introduced. Next, the sequence to sequence (S2S) architecture that is based on LSTM that is later used in the experiment is explored.

A. Long Short-Term Memory

RNNs can in principle use their feedback connections to store representations of recent input events. Usually this is done through Back-Propagation Through Time or Real-Time Recurrent Learning. These methods often fail, because the error signals that flow backward in time tend to blow up or vanish. LSTM [6] was designed to overcome this problem. It is overcome by an efficient gradient-based algorithm that enforces constant error flow through special units called memory cells, allowing it to store information for long periods of time. Each memory cell is built around a central linear unit with a self-recurrent connection called a self-loop (Figure 1). This self-loop, sometimes called forget gate, stores temporal information that is being stored in its state. There are also input gates and output gates which may use inputs from other memory cells to decide whether to store or to access certain information in its memory cell. The usage of these gates depends on the network topology. The following equations express the operations for a single LSTM cells c_j at time t

$$y^{in_j}(t) = f_{in_j}\big(net_{in_j}(t)\big) \tag{5}$$

$$y^{out_j}(t) = f_{out_j}\big(net_{out_j}(t)\big) \tag{6}$$

$$net_{in_j}(t) = \sum_u w_{in_j u} y^u(t-1) \tag{7}$$

$$net_{out_j}(t) = \sum_u w_{out_j u} y^u(t-1) \tag{8}$$

$$net_{c_j}(t) = \sum_u w_{c_j u} y^u(t-1) \tag{9}$$

$$y^{c_j}(t) = y^{out_j}(t) h\big(s_{c_j}(t)\big) \tag{10}$$

$$s_{c_j}(t) = s_{c_j}(t-1) + y^{in_j}(t) g\big(net_{c_j}(t)\big) \tag{11}$$

where u may stand for input units, gate units, memory cells, or conventional hidden units, y^{in_j}, y^{out_j}, y^{c_j} are activations corresponding to the input gate, the output gate, and the memory cell respectively, s_{c_j} is the internal state and f, g, h are appropriate activation functions.

B. Sequence to Sequence based LSTM

Instead of a simple RNN network with LSTM layers we introduce the architecture called sequence to sequence (S2S). S2S was introduced in [7] to map sequences of variable lengths in the field of language translation.

The S2S architecture consists of two LSTM networks, an encoder and a decoder (Figure 2). The encoder converts input sequences of variable lengths and maps them onto a vector of fixed length. This fixed length vector is then used as input to the decoder which then generates an output sequence of length n. Both are then trained to minimize a given loss of the target sequence given the input sequence.

The main advantage of the S2S architecture is that it allows sequences of variable lengths to be used as inputs to predict a future sequence of arbitrary length. The usage of the fixed vector enables it to learn the context of the data, which in our case is a function we want to approximate. For the differences between learning one step a time with ordinary RNN with LSTM and the S2S method see [2].

IV. Experimental Results

In the first part of this section we will introduce the neural network architecture used in our experiment. In the next parts, this section provides the experimental results obtained for the two models that were introduced.

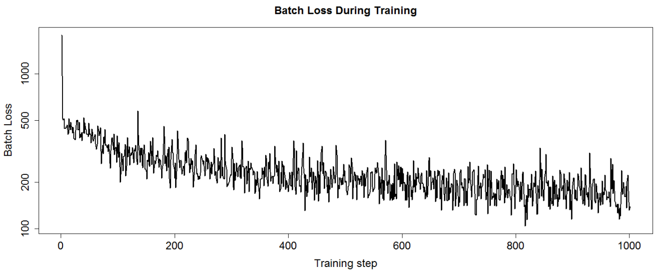


Fig. 3: The training batch loss for the neural network during experiment 1.

TABLE I: Loss on the validation dataset using normalized data, for the networks in both experiments and all 6 clusters in experiment 2.

Model	Loss value on the validation set
Aggregated model (Exp. num.1)	13150.32
Cluster 1 (Exp. num.2)	6025.25
Cluster 2 (Exp. num.2)	10720.85
Cluster 3 (Exp. num.2)	59930.37
Cluster 4 (Exp. num.2)	9328.46
Cluster 5 (Exp. num.2)	20550.43
Cluster 6 (Exp. num.2)	13548.95

TABLE II: The final MAPE on the test dataset for both experiments and all 6 clusters in experiment 2.

Model	MAPE [%]
Aggregated model (Exp. num.1)	**8.33**
Sum of models (Exp. num.2)	**7.06**
Cluster 1 (Exp. num.2)	5.38
Cluster 2 (Exp. num.2)	7.62
Cluster 3 (Exp. num.2)	28.16
Cluster 4 (Exp. num.2)	5.40
Cluster 5 (Exp. num.2)	6.72
Cluster 6 (Exp. num.2)	9.23

A. Network architecture used in the experiment

In our experiment, we used the S2S LSTM architecture to predict the future demand. Our neural network was composed of two layers of 40 LSTM hidden units in a S2S architecture. The input of our network consisted of a sequence of 96 time steps that represented the electricity load of 15 min. intervals. The output of the network was an estimation of the demand in the next 96 time steps. We use the mini-batch gradient descent [8] with RMSProp [9] as the gradient based optimizer to train the network. As we chose a small learning rate (=0.001), we perform 1000 training steps which corresponds to 1000 times showing a batch to the network.

During the training of the network and for testing when applying the computed model, the whole data was normalized with the mean μ and standard deviation σ of the training dataset.

As the loss function L to minimize during training we use the L$_2$ error with a L$_2$ regularization term (both without the square root) to get a better generalization capacity for the network [10] This function can be expressed by the following formula:

$$L = \sum_{t \in Batch} \|\boldsymbol{y}_{[t]} - \bar{\boldsymbol{y}}_{[t]}\|_2 + \alpha \|\boldsymbol{w}\|_2 \tag{12}$$

where $\boldsymbol{y}_{[t]}$ and $\bar{\boldsymbol{y}}_{[t]}$ are the vectors of 96 values of the actual and the predicted load corresponding to one step t of the batch being used, $\boldsymbol{w}$ are the weights of the system and α is the regularization strength, in our case 0.0003.

Since mini-batch gradient descent was used the training error fluctuates during learning (see Figure 3 for the training loss during the first experiment for the idea). To counter the fluctuating error, we used the validation dataset to find the loss on the whole validation dataset to determine the best model found during training.

To compare the precision of the forecasts of these methods, the forecast error of the electricity consumption was expressed by means of the MAPE (Mean Absolute Percentage Error). The MAPE is defined by the following formula:

$$MAPE = 100 \times \frac{1}{n} \sum_{i=1}^{n} \left| \frac{x_i - \bar{x}_i}{x_i} \right| \tag{13}$$

where x_i is the real consumption, $\bar{x}_i$ is the forecasted load and n is the length of the time series.

To determine the number of clusters during the second experiment, the Davies-Bouldin index [11] was used. The Davies-Bouldin index is a cluster separation measure which indicates the similarity of the computed clusters. The index uses the measure used in the clustering algorithm which is in our case the SBD. Let S_i be the dispersion measure inside the ith cluster and let $M_{i,j}$ be the measure of separation between cluster i and cluster j. In our case S_i is the average SBD distance inside the cluster and $M_{i,j}$ is the SBD between the centroids of cluster i and cluster j. The Davies-Vouldin index is then computed as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{S_i + S_j}{M_{i,j}} \tag{14}$$

where k is the number of clusters.

B. Experimental results using S2S without clustering

The first experiment that was performed was the prediction of the aggregated electricity demand of our whole network. The forecast was performed by training a neural network described in Section IV, with network parameters set as described in Section V. A. The network was trained on the training dataset which was from 01-07-2013 to 19-08-2014. See Figure 3 for the training error. For the final model, the best setting of the network was chosen such, that the loss on the whole validation dataset was the lowest (See table 1 for the final validation loss). The accuracy of the model was measured with the MAPE by taking the test dataset and predicting from the data for one day the whole following day. This is different to the validation and the training because this way during testing each value exists only once which is not the case during training and validation where each value of the time series exists 96 times (shifted by one time period between samples). The final MAPE can be found in Table 2,

TABLE III: The final MAPE on the test dataset for both experiments and all clusters in experiment 2.

k = Number of clusters	Davies-Bouldin index
2	1.51
3	2.96
4	2.62
5	2.38
6	**2.29**
7	2.33
8	2.34
9	3.47

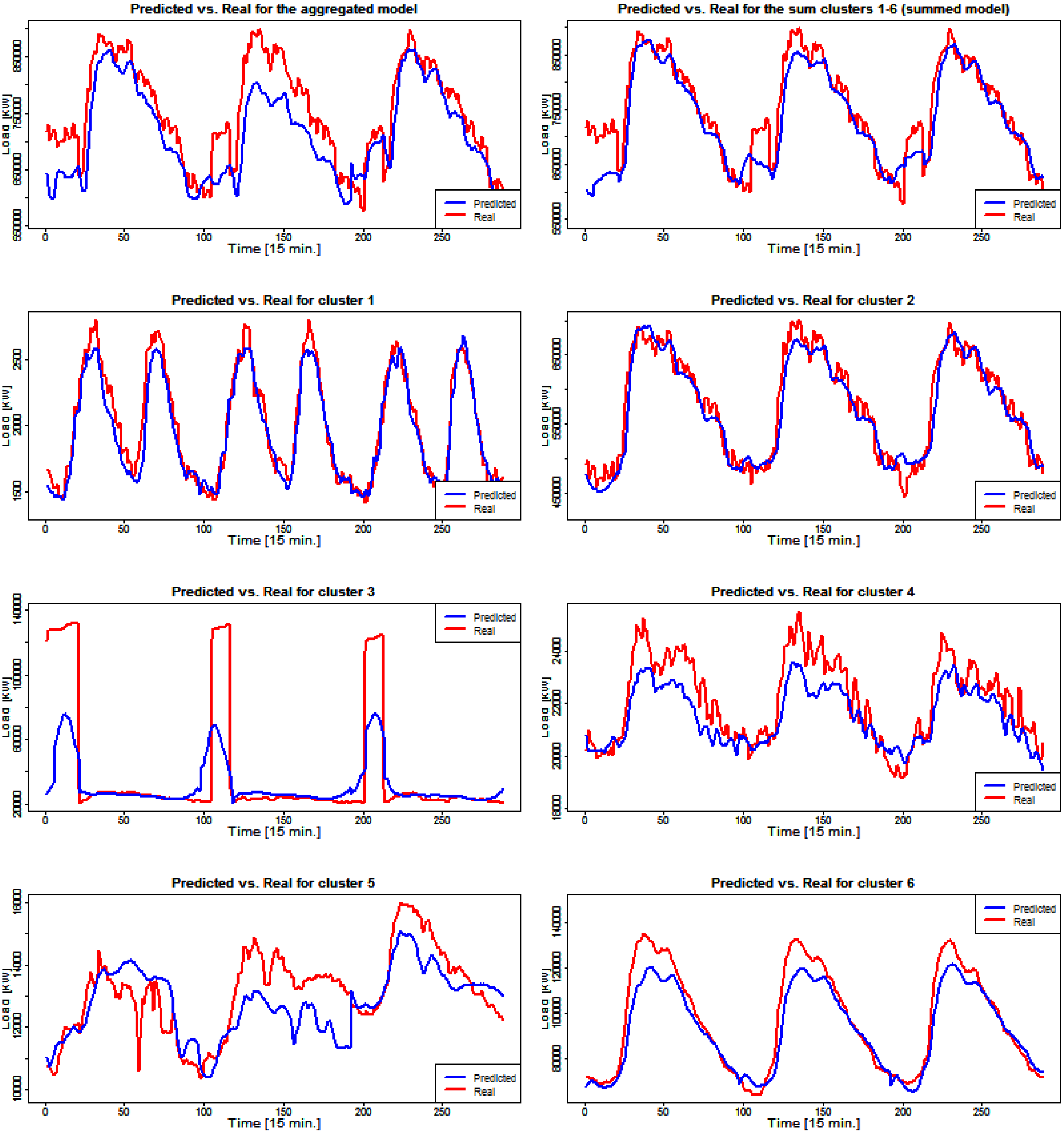


Fig. 4: The predicted (blue) and real (red) values of a part of the test dataset for all computed models in both experiments.

and in Figure 4 the actual load and the predictions of a part of the test dataset can be seen.

C. Experimental results using S2S with K-Shape clustering

The second experiment is consisted of first clustering the time series with the k-Shape algorithm (see Section II. C.). k-Shape clustering was done on the last 60 days of the training dataset on time series that were aggregated to the time period of one hour. The number of clusters k was determined by using the Davies-Bouldin index. We chose $k = 6$ because it is a local minimum of the index (See Table 3 for details).

For each cluster, we constructed the same kind of neural network as in the first experiment, with the same network parameters. Normalization of the data during training was done with the mean and standard deviation of the training data of each cluster separately. After training the network, we chose the best setting for each network with the validation datasets. For the final aggregate predictions, each cluster predicted the load separately and then the sum of these loads was taken.

The resulting MAPE for each cluster and for the aggregated model can be seen in Table 2. We can see that the clustering approach reduced the MAPE by 15% compared to the approach without clustering. In Figure 4 the predictions with the actual load can be seen for each cluster and for the aggregated models for a part of the test dataset. All predictions demonstrate a bias by underestimating the actual load. This is caused by an increasing electricity demand between the training and test set of the data. We can see that the prediction for cluster 3 is not good (this can also be seen from Table 2). This is caused by the regularization loss from Equation 12, with makes the curve smoother, then the data in cluster 3. Individual setting of the parameters of the network could counter this problem.

V. Conclusion

The goal of the presented work was to investigate the need for clustering when using LSTM based neural networks with S2S architecture for grid level energy load forecasting on real world data. The paper described two approaches used for predicting the aggregated load demand using LSTM S2S based neural networks. The first method was without clustering (the simple aggregation of the consumers) and the second method used *k*-Shape clustering. Using the second method the neural network could predict the aggregated load more accurately, than the method without clustering. When using clustering, the neural network model had to predict curves that were easier to predict when using slight generalization in the neural network architecture. The paper did not use any assumption on the data such as cyclic (or seasonal and trend) behavior making the results applicable to other field than energy load forecasting.

Acknowlegment

This work was partially supported by the Scientific Grant Agency of The Slovak Republic, Grant No. VG 1/0752/14.

References

[1] P. Laurinec and M. Luck, “Comparison of Representations of Time Series for Clustering Smart Meter Data,“ Proceedings of the World Congress on Engineering and Computer Science, vol. 1, 2016.

[2] D. L. Marino, K. Amarasinghe and M. Manic, “Building Energy Load Forecasting using Deep Neural Networks,“ in 42nd Annual Conference of the IEEE Industrial Electronics Society, 2016.

[3] S. Bandyopadhyay, T. Ganu, H. Khadilkar and V. Arya, “Individual and Aggregate Electrical Load Forecasting: One for All and All for One,” in e-Energy ’15 Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems, New York, NY, USA, 2015.

[4] F. Fahiman, S. M. Erfani, S. Rajasegarar, M. Palaniswami and C. Leckie, ”Improving Load Forecasting Based on Deep Learning and K-shape Clustering,” in 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017.

[5] J. Paparrizos and L. Gravano, “Series, k-Shape: Efficient and Accurate Clustering of Time,“ in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015.

[6] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,“ Neural Computation, vol. 9, pp. 1735-1780, 15 November 1997.

[7] I. Sutskever, O. Vinyals and Q. V. Le, “Sequence to sequence learning with neural networks,“ in NIPS’14 Proceedings of the 27th International Conference on Neural Information Processing Systems , Montreal, Canada, 2014.

[8] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,“ in conference paper at ICLR 2017, 2016.

[9] T. Tieleman and G. H. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,“ COURSERA: Neural networks for machine learning, vol. 4, no. 2, pp. 26-31, 10 2012.

[10] M. T. Hagan, H. B. Demuth, M. H. Beale and O. D. Jess, “Neural Network Design“, Boston, PWS Publishing Co..

[11] D. L. Davies, D. W. Bouldin, “A cluster separation measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1, no.2, pp.224-227, 1979.

Clustering analysis of phonetic and text feature vectors

Milan Jičínský
University of Pardubice
Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
milan.jicinsky@student.upce.cz

Jaroslav Marek
University of Pardubice
Faculty of Electrical Engineering and Informatics
Studentská 95, 530 02 Pardubice I, Czech Republic
jaroslav.marek@upce.cz

***Abstract*— Our goal is to show an example of using statistical methods to analyse some attributes of speeches. For this purpose, the New Year's Day speeches of Czech and Czechoslovak presidents are chosen. The aim of our study is researching similarities among these speeches and their recognizability through the history of Czechoslovak politics. All presidents are compared between each other. The comparison method is based on principal component analysis and cluster analysis. Important part is creating a feature vector. The feature vector doesn't have to be the same for successful clustering. There are many varieties and combinations of features that can be selected and used. Correlated variables must be discarded. The most significant features are chosen to represent and characterize the speaker. Some speakers can have something in common according to the chosen features. Or on the other hand they can differ much more from others. This kind of approach can help us to recognize a speech pattern of each spokesman independently.**

Keywords—clustering; New Year's Day speeches; President; feature vectors; voice analysis; energy; zero crossing rate; speech velocity; linguistics; phonetics; segmentation; frames; audio processing; speaker comparison; principal component analysis; cluster analysis

I. Introduction

Clustering is a very spread statistical method used in various field of research [1-3]. Well described fundamentals and clustering algorithms can be find in [4]. Applying this method even for audio files including a music [5] and voice recordings isn't an exception. Another example of using clustering in this case for segmentation and classification of audio files is well described in [6]. Comparing audio sets can be very efficient but it's important to know that only sampled recording or voice fragments cannot be a subject of clustering. It's because of representation of speech as a raw signal. This means e.g. millions of sampled values. Those values depend on time and content of speech. Those values have no use for us. It's necessary to make some audio processing first. Instead of using signal samples, each recording is represented by parameters also called features. These features can be put together to create a feature vector. Only the representation of recording as a feature vector is acceptable for further statistical analysis.

In this research, our attention has been given to find some similarities of New Year's Day speeches by using statistical methods. This has been chosen example of showing how to extract information from available data and using them for researching speaker similarities considering the text and speech itself. The speech of Czech and Czechoslovak presidents can be characterized by various voice characteristics such as zero crossing rate, log energy, speech velocity, spectral energy and many more. To reduce the high-dimensional data, the principal component analysis and the hierarchical cluster analysis will be used. We will take a look at possibilities of using statistical method, phonetic analysis and mathematical linguistics for comparing of political speeches. The main goal is to show how recordings can be analysed and compared among each other in different way than we are used to. Different scientific approach can be achieved just by linking of linguistics, phonetics and statistical methods using clustering algorithms.

By help of linguistic characteristics and voice characteristics of speakers the similarities of New Year's Day Speeches of Czech presidents can be measured. The next step is to explore the differences between the result clusters. These results can show us the partial influences of speaker's characteristics. Our aim is to obtain the best results using and combining only other phonetic and text based features.

II. SAMPLE OF SPEECHES

A. Data

The data reserved for our research come from [7]. This is web audio archive containing almost every single presidential speech since 1935. The sources of these speeches and their transcriptions are archive of president's office, linguist Jaroslav David and Moravian Library. But thanks to Český rozhlas everything is in one place.

Each speech was recorded separately and sampled at 16 kHz using Audacity software. As mentioned before the recording must be edited. Otherwise it couldn't be used in our case. First, it is necessary to get rid of all parts containing a music, long silence or even a voice of moderator who has an introduction speech at the very beginning of recording. Finally, the data are ready to be examined.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

B. *Measuring of voice parameters*

Since we have the recordings edited every speech should be segmented into smaller parts called frames. The frames are typically 20ms long and they are overlapping each other right in the half. Segmentation is followed by parameterization step. During the parameterization, the features are evaluated for each frame. These features can be divided into some groups. We can distinguish basic, spectral and cepstral features. Each of those groups can be considered either static or dynamic. Static features are computed exactly per the corresponding formulas. They must be calculated before the dynamic features. It's because dynamic features are given by static ones. While the static parameters of feature vector have their own meaning in sense of signal analysis and they are related to frequency and other measurable parameters, the dynamic features only express changes of the static values among frames. In the various publications, the dynamic features are also called delta features because it can be considered as the mathematical derivative of original parameter. Expression "delta" is used for the first derivatives. The second derivatives are called "delta-delta”. The example of measured values for chosen speech is given in Table I. These numbers are the result of analysis for Masaryk's speech dated 1935 (left column) and Havel’s speech dated 1996 (right column).

TABLE I. AN EXAMPLE OF SPEECH FEATURE VALUES

Feature type		**Speech feature vector**		
		Feature name	***Values***	
Static	*Basic*	Speech velocity	1.335	1.961
		E – energy	9.333	8.573
	Spectral	Bk – bin 0 – 500 Hz	11.283	10.257
		Bk – bin 500 – 1000 Hz	9.263	8.946
		Bk – bin 1 – 1,5 kHz	8.816	8.812
		Bk – bin 1,5 – 2 kHz	8.138	8.416
		Bk – bin 2 – 2,5 kHz	7.947	8.009
		Bk – bin 2,5 – 3 kHz	8.056	7.837
		Bk – bin 3 – 3,5 kHz	8.220	7.832
		Bk – bin 3,5 – 4 kHz	8.143	7.315
Dynamic (delta)	*Basic*	|ΔE| – delta energy	0.245	0.363
		|ΔΔE| – delta-delta energy	0.346	0.461
	Spectral	|ΔBk| – bin 0 – 500 Hz	0.294	0.413
		|ΔBk| – bin 500 – 1000 Hz	0.290	0.475
		|ΔBk| – bin 1 – 1,5 kHz	0.255	0.492
		|ΔBk| – bin 1,5 – 2 kHz	0.214	0.492
		|ΔBk| – bin 2 – 2,5 kHz	0.161	0.467
		|ΔBk| – bin 2,5 – 3 kHz	0.162	0.465
		|ΔBk| – bin 3 – 3,5 kHz	0.171	0.479
		|ΔBk| – bin 3,5 – 4 kHz	0.200	0.479

[a] Source: own.

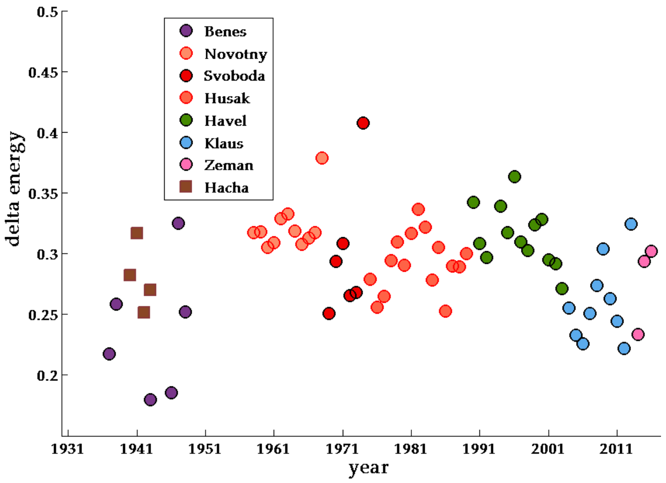


Fig. 1. Delta energy representing voice parameters. Source: own.

While energy and speech velocity were shown in [8], first and second derivative of energy values are given by Fig. 1 and Fig. 2. As can be seen a range of chosen feature values is completely different. Speech velocity depends on pace of the speaker. The number shows how many words were said within one second. Typical range is from 0,9 to 2,5. Average energy ranges between 7 and 10. As for spectral energy *Bk*, the situation is very similar but it differs depending on which bin was used. On the other hand, however delta features ordinarily have very low values. This is caused by the fact that the differences between frames are very small positive or negative numbers and they are even approaching to the zero value. This is the reason why original delta features were not used. Because their mean value is almost equal to zero. Instead of it the mean value was calculated from absolute value of those features. This is the only way how to use these dynamic parameters for clustering and it can be considered as a slightly different scientific approach. Finally, we decided not to include a Zero crossing rate due to the very high variance of values within one president. This variance is also shown in [8].

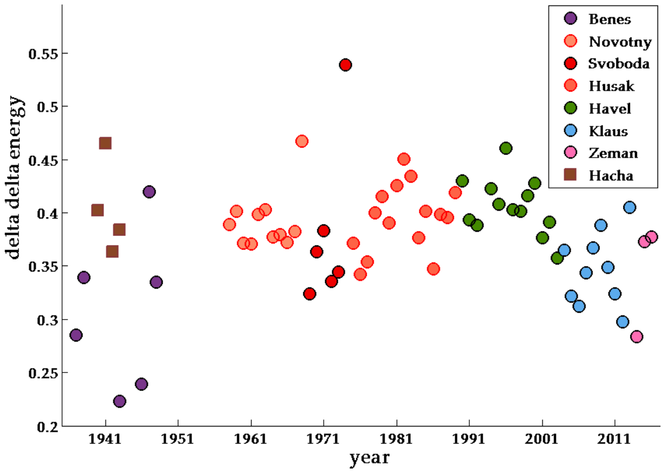


Fig. 2. Delta delta energy representing voice parameters. Source: own.

C. *Text feature extraction*

Some text related features can be defined and used as feature vector the same way as the phonetic parameters. Special feature vector representing some text characteristics were designed for the purposes of the article. Some of those features were already discussed in [8]. Total number of words has a relation with length of the whole speech. An amount of different words is the result of saying the same thing in different ways or just simply not intending to repeat the same words. Even sort of language richness can be related to this parameter. Even length of words plays an important role. While conjunctions and prepositions doesn't vary too much, the most contained word can easily characterize the speaker. Table II is an example of text feature vector. It's organized the same way as previous table. It means that the left column is reserved for Masaryk and the right one for Havel. The dates of chosen speeches remain the same.

All speech processing has already been done in [8]. Even more details about previous steps can be find there too. Fundamentals of audio processing, segmentation and parameterization are included. Unfortunately, only basic static features were used and described (energy, speech velocity, zero crossing rate). The rest of features will be named and shown in upcoming yet unpublished paper [9].

TABLE II. AN EXAMPLE OF TEXT FEATURE VALUES

Feature type		**Text based feature vector**		
		Feature name	***Values***	
TEXT	*Numbers*	Total number of used words	259	2749
		Amount of different words	190	1390
		Mean length of words	5.514	5.206
		Most dominant length of words	7	2
	Words	The most contained word	nation	country
		Most used conjunctions and prepositions	and	and

b. Source: own.

III. THE CLUSTERING

A. *The Principal Component Analysis*

Some correlation between the voice characteristics of the speeches occurred, and that's why it is better to use principal components analysis.

In the Principal Components Analysis (PCA), the data are summarized as a linear combination of an orthonormal set of the vectors. The first principal component accounts for as much of the variability in the data as possible, and each successive component represents as much of the remaining variability as possible. This is the same as performing the singular value decomposition of the covariance matrix var **X** = **U D V,** where **D** is diagonal matrix of eigen-values, and **U**, **V** are orthonormal. Cf. [1].

The results of PCA are shown in Fig. 3-7. We use up to three principal components.

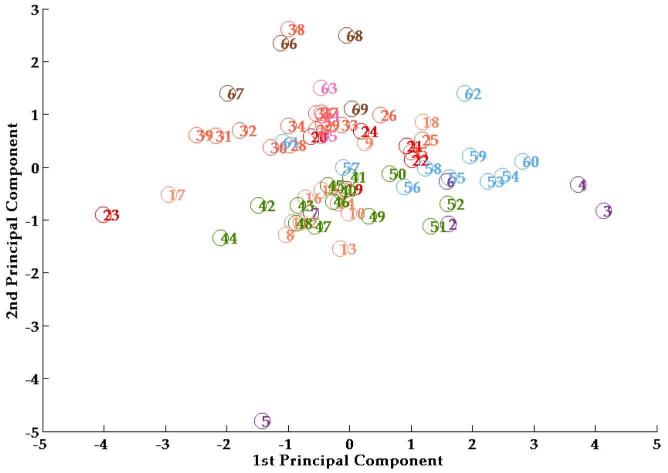


Fig. 3. PCA using 4 speech features. Source: own.

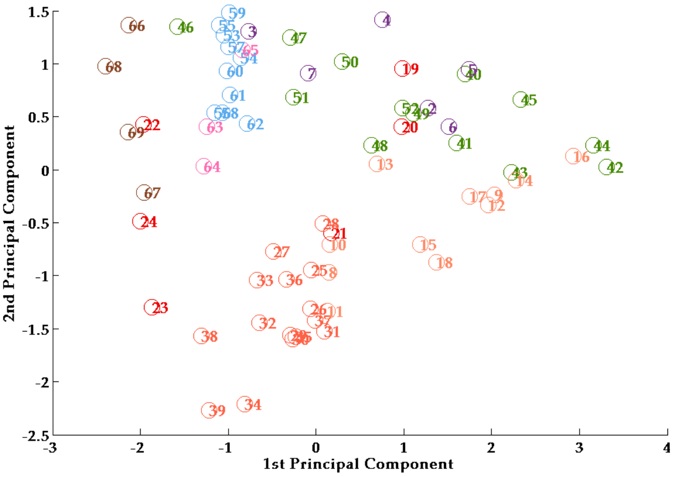


Fig. 4. PCA using 3 text features. Source: own.

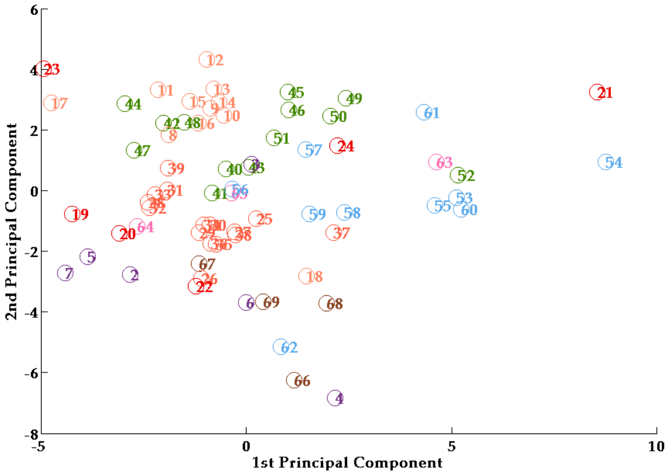


Fig. 5. PCA using 20 speech features. Source: own.

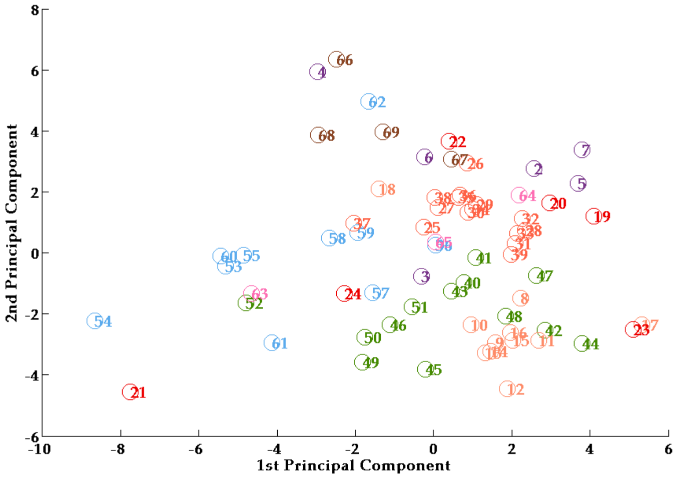


Fig. 6. PCA using 4 speech and 3 text features. Source: own.

It is necessary to know that results of the clustering may differ depending on chosen feature vector. That is why designing the vector can be considered as the most important part. More combinations were tried during our research. The partial component analysis was realized firstly for four parameters representing only speech (voice) - average energy, speech velocity, delta energy and delta-delta energy. The results can be seen at Fig. 3. Then we came up with idea of extracting only text based features. So, the second experiment were made using three representative linguistic parameters - total number of words, amount of different words, average length of words. Graphical interpretation is at Fig. 4. Then we tried to use the same feature vector as defined in Table I. Results of PCA for this vector containing twenty parameters are presented at Fig. 5. After these three tries we wanted to combine both linguistic and phonetic characteristics. And so, the next feature vector contains original twenty values of voice parameters plus three text based. This is shown in Fig. 7. The feature vector containing the fewest number of parameters but still combining text and voice parameters is used for PCA at Fig. 6.

The proportions of the principal component on the total variability of the original data for the different features are shown in the Table III.

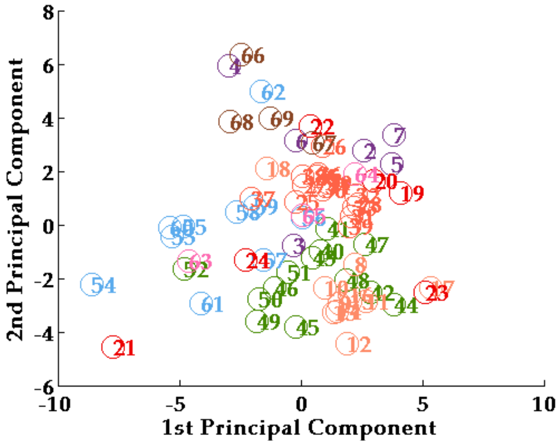


Fig. 7. PCA using 20 speech features and 3 text features. Source: own.

TABLE III. RESULTS OF PCA

Total variability	First component	First and second component	First, second and third component
Data20+0	55.6 %		
Data0+3	66.5 %	99.7 %	
Data4+0	40.3 %	71.1 %	82.3 %
Data4+3	35.5 %	63.4 %	75.2 %

c. Source: own.

We can see that the largest value of proportion on total variability occurs for the features data0+3. Thus, it is clear, that the author of the text is better defined by the text characteristics than the speaker described by the speech characteristics.

However, we will use the data4 file for further clustering, which is less suitable for criterion of total variability proportion maximization. The reason why the data 4+3 was used is an effort to work with both types of characteristics.

By help of another neighbour algorithm and the principal components we tried to find the clusters of chosen speeches.

The results obtained by using the hierarchical clustering are given in Table IV.

B. Cluster Analysis

In the table, we calculate the measure of membership into clusters for all speeches according to formula

$$\mu\left(x_i, v_j\right) = 1 - \frac{d\left(x_i, v_j\right)}{\max d\left(x_i, v_r\right)}, i = 1, \ldots, n; r = 1, \ldots, R\,, \qquad (1)$$

where d is the Euclidian distance, R is the number of patter objects. The most significant measure of membership is bold.

As for the Table IV, C1, ..., C8 are designations for patterns defined as centre of gravity of the first three components of PCA realized for 8 presidents and obtained for data containing 4 voice features and 3 text based. There are only 8 pattern categories because we have 8 presidents. Masaryk, Gottwald and Zápotocký are removed from further statistical analysis due to the lack of data. Each of them has only one short recording available. Discarding them makes the PCA results better. Each president should fit into the category with the same number as his ranking in the table. That means Beneš belongs to C1, Novotný to C2 and so on.

The only one whose classification is 100% accurate is the president of protectorate Hácha. He had significantly lower energy during his speeches. Havel has the only one wrongly classified speech within year 2003. Beneš and Novotný are fitting the right patterns too. As for those presidents, we can be satisfied with the final score. Results provide us an information that these presidents were very specific and they can be easily recognized and separated from the rest. They wanted their speeches to be the reflections of their own opinions.

TABLE IV. RESULTS OF CLUSTERING

president	C1	C2	C3	C4	C5	C6	C7	C8
Beneš 37	**26%**	9%	16%	12%	14%	11%	12%	0%
Beneš 38	**29%**	9%	17%	17%	10%	0%	12%	7%
Beneš 43	**20%**	16%	14%	10%	18%	13%	9%	0%
Beneš 46	**27%**	0%	12%	12%	4%	12%	17%	16%
Beneš 47	18%	12%	18%	**20%**	11%	0%	13%	9%
Beneš 48	**25%**	0%	15%	16%	5%	6%	19%	14%
Novotný 58	18%	13%	17%	**19%**	11%	0%	12%	10%
Novotný 59	10%	**26%**	16%	13%	22%	5%	8%	0%
Novotný 60	7%	**22%**	17%	13%	24%	7%	9%	0%
Novotný 61	7%	**22%**	17%	13%	25%	8%	9%	0%
Novotný 62	9%	**27%**	15%	13%	22%	6%	8%	0%
Novotný 63	7%	**24%**	15%	12%	23%	9%	8%	0%
Novotný 64	10%	**24%**	15%	12%	22%	9%	8%	0%
Novotný 65	8%	**24%**	16%	12%	24%	8%	8%	0%
Novotný 66	10%	**25%**	15%	12%	22%	8%	8%	0%
Novotný 67	9%	**25%**	16%	13%	23%	7%	8%	0%
Novotný 68	8%	22%	17%	13%	**24%**	7%	9%	0%
Svoboda 69	21%	0%	15%	14%	6%	12%	**20%**	11%
Svoboda 70	19%	19%	18%	17%	16%	0%	9%	3%
Svoboda 71	16%	13%	19%	20%	13%	0%	12%	8%
Svoboda 72	0%	4%	12%	9%	14%	**36%**	17%	8%
Svoboda 73	19%	0%	18%	22%	4%	0%	20%	18%
Svoboda 74	3%	**26%**	19%	19%	22%	0%	9%	2%
Husák 75	2%	10%	18%	14%	19%	**21%**	15%	0%
Husák 76	14%	3%	**27%**	23%	11%	3%	20%	0%
Husák 77	18%	2%	19%	**22%**	5%	0%	19%	15%
Husák 78	6%	2%	24%	**30%**	8%	0%	21%	9%
Husák 79	12%	7%	22%	**25%**	10%	0%	16%	8%
Husák 80	6%	7%	22%	**28%**	10%	0%	17%	10%
Husák 81	5%	13%	22%	**24%**	14%	0%	14%	8%
Husák 82	7%	12%	21%	**24%**	13%	0%	14%	9%
Husák 83	10%	15%	21%	**22%**	15%	0%	12%	4%
Husák 84	8%	7%	22%	**28%**	10%	0%	17%	9%
Husák 85	19%	4%	22%	**22%**	9%	0%	17%	6%
Husák 86	10%	4%	21%	**27%**	8%	0%	18%	11%
Husák 87	5%	0%	16%	17%	7%	14%	**25%**	17%
Husák 88	2%	0%	20%	26%	5%	1%	**24%**	22%
Husák 89	5%	21%	**22%**	21%	21%	0%	10%	0%
Havel 90	10%	18%	19%	15%	**22%**	6%	10%	0%
Havel 91	14%	19%	18%	15%	**20%**	5%	9%	0%
Havel 92	8%	27%	16%	14%	**22%**	5%	8%	0%
Havel 94	10%	19%	17%	13%	**23%**	8%	9%	0%
Havel 95	9%	27%	16%	14%	**22%**	5%	8%	0%
Havel 96	4%	16%	17%	13%	**22%**	16%	12%	0%
Havel 97	6%	17%	17%	12%	**23%**	15%	11%	0%
Havel 98	11%	24%	18%	15%	**21%**	3%	8%	0%
Havel 99	9%	25%	16%	13%	**23%**	6%	8%	0%
Havel 00	7%	15%	15%	11%	**21%**	20%	11%	0%
Havel 01	6%	13%	16%	11%	**21%**	20%	12%	0%
Havel 02	12%	17%	16%	11%	**21%**	13%	10%	0%
Havel 03	6%	7%	22%	**28%**	10%	0%	17%	10%
Klaus 04	5%	13%	22%	**24%**	14%	0%	14%	8%
Klaus 05	7%	12%	21%	**24%**	13%	0%	14%	9%
Klaus 06	10%	15%	21%	**22%**	15%	0%	12%	4%
Klaus 07	8%	7%	22%	**28%**	10%	0%	17%	9%
Klaus 08	19%	4%	22%	**22%**	9%	0%	17%	6%
Klaus 09	10%	4%	21%	**27%**	8%	0%	18%	11%
Klaus 10	5%	0%	16%	17%	7%	14%	**25%**	17%
Klaus 11	2%	0%	20%	**26%**	5%	1%	24%	22%
Klaus 12	5%	21%	**22%**	21%	21%	0%	10%	0%
Klaus 13	10%	18%	19%	15%	22%	6%	10%	0%
Zeman 14	0%	8%	15%	12%	16%	**27%**	16%	6%
Zeman 15	18%	0%	13%	15%	4%	9%	20%	**22%**
Zeman 16	0%	2%	14%	13%	11%	**28%**	20%	12%
Zeman 17	12%	10%	20%	**23%**	11%	0%	14%	8%
Hácha 40	11%	0%	12%	15%	4%	9%	20%	**29%**
Hácha 41	5%	0%	18%	24%	4%	0%	22%	**26%**
Hácha 42	9%	0%	12%	15%	4%	12%	21%	**26%**
Hácha 43	12%	0%	14%	16%	4%	8%	21%	**24%**

d. Source: own.

On the other hand, Svoboda, Klaus and Zeman are hard to distinguish. They have zero positive hits according to Table IV. Absolutely the worst classification can be recognized at speeches of president Klaus. His speeches are the most like president Husák.

IV. Alternative Possible Solution

Commonly used methods of comparing speeches are basically based on cepstral features and Hidden Markov Models. The clustering can be made using Mel-Frequency Cepstrum Coefficients (MFCC) and Linear Prediction Cepstral Coefficients (LPCC). Alternatively, even more efficient methods (especially for classification of speaker) use Gaussian Mixture Models, artificial intelligence (neural networks) and the most recent method called i-vectors. As for recognition of speaker it is very popular to create robust and text-independent recognition systems nowadays. Dynamic Time Warping (DTW) and HMM can't be used for this purpose. Our intention was to compare president speeches not the biometric recognition and classification of speaker. This is the reason why an advanced techniques and methods were intentionally omitted. More information about using HMM provides [10]. So, the alternative way can be realized by researching only recordings and omitting the transcript analysis.

In this case the gist of article is to search for similarities in president speeches according to the recordings and transcripts. Feature vectors were created for this purpose as noted above in previous chapter. These feature vectors combine the most important characteristics of speeches and text. They can be used for separated clustering or for linking text and speech together. Many more combinations of features may exist. This leads to the opportunity of further research in this field of study.

V. Concluding Remarks

Unfortunately, the dendrogram isn't suitable for graphical interpretation of clustering due to the number of president speeches. Using the feature vector containing twenty features as shown above doesn't bring expected results. This is influenced by correlation among most of these parameters. Even reducing the number of features to only four (average energy, speech velocity, delta energy and delta-delta energy) doesn't help that much to detect the speaker. The combination of text features and most significant phonetic features leads to the best results. Adding the cepstral features would be probably the best option for improving the results and then it could end up better. But thanks to the Table IV, the probabilities of belonging to the right cluster are relatively high. It is still very efficient even if some speeches of different presidents were clustered together. According to that fact we can find presidents with low rate of individualism expressed in the speeches – Zeman, Klaus and Svoboda. The rest can be more easily distinguished by their attributes that differ from one another. This research also proves that cepstral features are very hard to replace. Using cepstral features, Hidden Markov Models, Gaussian Mixture Models and neural networks may still provide better results, but doesn't allow to combine these features the way we did or they are not using any features at all.

ACKNOWLEDGMENT

This research was supported by the Internal Grant Agency of University of Pardubice, the project SGS 2017 024.

REFERENCES

[1] Steinbach, Michael, et al. "A comparison of document clustering techniques." In: KDD workshop on text mining. 2000. p. 525-526.

[2] Ruspini, Enrique H. "Numerical methods for fuzzy clustering." *Information Sciences*, 1970, 2.3: 319-350.

[3] Vesanto, Juha; Alhoniemi, Esa. "Clustering of the self-organizing map." IEEE Transactions on neural networks, May 2000, 11.3: 586-600.

[4] Jain, Anil K.; Dubes, Richard C. "Algorithms for clustering data." Prentice-Hall, Inc., 1988.

[5] Levy, Mark; Sandler, Mark. "Structural segmentation of musical audio by constrained clustering." IEEE Transactions on Audio, Speech, and Language Processing, January 2008, 16.2: 318-326.

[6] Lu, Lie; Zhang, Hong-Jiang; Jiang, Hao. "Content analysis for audio classification and segmentation." IEEE Transactions on speech and audio processing, December 2002, 10.7: 504-516.

[7] URL: http://www.rozhlas.cz/zpravy/data/_zprava/od-tgm-k-zemanovi-poslechnete-si-vanocni-a-novorocni-projevy-vsech-prezidentu--1436738. [accessed. 2017-08-13].

[8] Jičínský, Milan; Marek, Jaroslav. "New Year's Day speeches of Czech presidents: phonetic analysis and text analysis." In: IFIP International Conference on Computer Information Systems and Industrial Management. Springer, Cham, May 2017. p. 110-121.

[9] Jičínský, Milan, "Features reserved for clustering of New Year's Day speeches of Czech presidents," unpublished.

[10] Abdallah, Sayed Jaafer; Osman, Izzeldin Mohamed; Mustafa, Mohamed Elhafiz. "Text-independent speaker identification using hidden Markov model." World of Computer Science and Information Technology Journal, 2012, 2.6: 203-208.

Towards a Uniform Code Annotation Approach with Configurable Annotation Granularity

Ján Juhár and Liberios Vokorokos
Department of Computers and Informatics
Technical University of Košice, Slovakia
{jan.juhar, liberios.vokorokos}@tuke.sk

***Abstract*—Transforming software system requirements and design into a code leads to loss of high-level details, complicating the program comprehension. This loss of details can be partially prevented by preserving concerns within code annotations. However, each annotation type has its limitations and the optimal annotation granularity levels may vary for different codebases. In this paper we present a concept for a code annotation tool that unifies workflows of the individual annotation types, making it possible to use the most fitting one per annotation use case with the same user interactions. Moreover, our concept supports changing annotation granularity levels by specifying tree patterns for annotable code elements. Our goal is to augment integrated development environments with concern-aware views to facilitate the comprehension process.**

I. Introduction

Program comprehension tasks are often complemented by Integrated Development Environments (IDEs) [1]. These environments integrate a wide variety of tools that programmers can use to quickly navigate the source code and to uncover relations of its elements even in large codebases. However, operations these tools are able to perform are usually limited by the amount of information retrievable from the code itself; by its *intrinsic metadata* [2] stemming from the implementation language semantics or coding conventions. This restricts the tools to work within the low-level *solution domain*. The *problem domain*, consisting of high-level functional and non-functional specifications, descriptions of user-facing features, or other software concerns, is separated by the *semantic gap* [3]. This gap is created during system implementation by *lossy* transformation of the problem domain to the solution domain. That makes understanding the original intent behind a piece of code represents a significant problem in software development [4].

Due to the semantic gap, tool support for bridging solution domain with the problem domain needs additional code-related metadata that will capture higher-level details. In our work in the area of program comprehension we focus on advancing IDE capabilities for working with custom code metadata related to high-level concerns. Such metadata, or *concern annotations*, if properly leveraged by IDE tools, may help in the comprehension process. In this paper we present our motivation for and progress towards a code annotation tool that will allow us to augment IDE tools with concern-related metadata.

In section II-B we discuss possible ways of annotating the code; the positive and negative properties of individual annotation types. We also deal with the topic of appropriate granularity level needed to preserve concerns with code annotations (section III). In section IV we present a concept of a code annotation tool with uniform annotation approach covering multiple annotation types and allowing to configure annotation granularity levels. A prototype implementation of the concept, created as an IDE plugin, will serve as a basis for augmenting IDE tools with concern metadata and for evaluations of effects such tool will have on program comprehension.

II. Motivation

The existence of the semantic gap stimulated creation of various methods intended to retrieve information otherwise lost or scattered in the code. These methods shaped up the two following general approaches:

- *recovery* of the information from the code by means of reverse engineering, and
- *preservation* of programmer's thoughts in software artifacts.

For feature location tools, belonging under the recovering approach, it is common to produce lists of files, classes, methods, or statements relevant to a feature [5]. Preserving approaches, e.g., leveraging language-level code annotations [6], directly assign high-level information to code elements. In both cases, the retrieved information represent metadata of code elements.

A. Need of Concern Annotations

Some concerns, however, may be too dissolved within the code to be recovered by methods of reverse engineering. This is true especially for design decisions that do not have clearly established conventions (e.g., naming conventions) that would help to identify them in the code. O the other hand, methods that proactively preserve high-level concerns during edit-time have an advantage of being able to significantly narrow the semantic gap [3].

Consider, for example, a method of concern annotation through language-level annotations (LLAs) [6], an usage of which is shown on Java annotations in listing 1. The annotations `@NoteChange` and `@TagManagement` are used to explicitly convey concerns implemented by the annotated

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Listing 1. Concerns captured by Java annotations

```java
@NoteChange
@TagManagement
public void addTag(String tag) {
  // ...
}
```

method for rapid building of a reader's mental model of the implementation.

Of course, such explicit recording of the mental model comes with additional cost: programmers must spend some extra time to record it. However, we believe that appropriate granularity of preserved details combined with effective method of preserving and presenting them could keep the benefits of more accurately described concerns in code annotations above this cost.

B. Limitations of Concern Annotations

Multiple approaches can be taken in order to annotate code with some sort of metadata (in our case with high-level concerns). When annotating code, a programmer can choose from the following annotation types:

- *Language-level annotations*, using native language constructs for metadata.
- *Structured code comments*, allowing to parse metadata from comments.
- *External*[1] *annotations*, using a supporting tool to assign metadata to arbitrary code fragments.

Each of these types has different limitations and fits different use cases. The limitations are shown in relation to each other in fig. 1 and are discussed below.

We introduced the concept of LLAs in section II-A. The advantage of such approach is that annotations become a part of the code and can be processed by standard language tools and IDEs. Thus the annotations can be effectively used (e.g., with *Find usages* IDE feature) and managed (e.g., through refactoring). But, depending on the language features, LLAs are available either to a fixed set of annotable code elements, or not at all.

If the LLAs are not applicable, they can be replaced with specially structured comments. Usually, the comments take form of a simple domain-specific language (DSL), e.g., reuse the syntax of *JavaDoc* comments [7]. An example of such comment is shown in listing 2, capturing the same concerns as were captured by LLAs in listing 1.

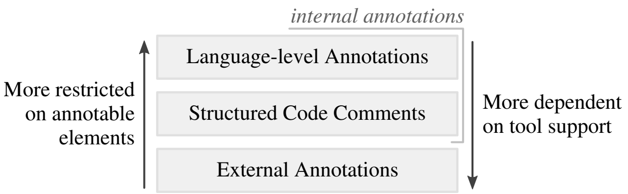
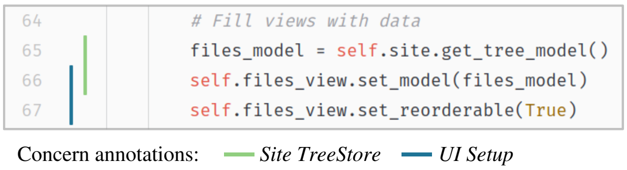

internal annotations
Language-level Annotations
Structured Code Comments
External Annotations
More restricted on annotable elements
More dependent on tool support

Fig. 1. Limitations of concern annotation types

Listing 2. Concerns captured in a structured comment

```java
// @concerns: NoteChange, TagManagement
public void addTag(String tag) {
  // ...
}
```



```python
        # Fill views with data
        files_model = self.site.get_tree_model()
        self.files_view.set_model(files_model)
        self.files_view.set_reorderable(True)
```

Fig. 2. External concern annotations shown in the editor gutter

Despite having formal syntax, comment annotations are somewhat less formal than LLAs because the custom syntax is not understood by the hosting language tools, and it may not always be clear to which code element a comment pertains. The syntax of the comments may also introduce unwanted "noise" to the code. On the plus side, comments are significantly less restricted on where they can be placed in the code compared to the LLAs.

Still another approach is taken with annotations stored externally to the code and managed by a supporting tool (e.g., as does *ConcernMapper* [8] or the tool from our previous work [9] shown in fig. 2). Such annotations are effectively superimposed over the code, which can be annotated without being changed. The annotation placement can be, at least theoretically, completely arbitrary.

The negative side is maintainability of such annotations. Their locations need to be kept in sync with changes made to the code they pertain to. Usually it is possible to achieve this while the code is being edited with the awareness of the supporting tool by tracking positions of annotated code fragments; problematic are changes made without the annotating tool. This problem cannot be fully prevented, and thus the usage of such annotations should be limited to cases where other types cannot be used.

The discussed limitations show that no single annotation type can be universally considered for the "best" one. However, usual shortcoming of tools working with concern annotations is that they support only one type (partial exception is the *TagSEA* tool [7], see section V) and thus cannot always be used. But tool supporting all these concern annotation types could use the appropriate one depending on a particular use case, with a uniform annotation process for all annotation types.

III. On Annotation Granularity Levels

In our work we focus on annotations for preservation of high-level concerns. Considering this use case, the code

[1] Complementing the *external* annotation type, the other two can be considered for *internal* annotations because they are made persistent part of the code.

element granularity levels that can be annotated represent an important aspect needed to be considered. However, granularity commonly used in existing tools related to concern annotations does not go below the declaration level of class members (see overview of such tools in section V). Will the cost of preserving and maintaining a finer concern granularity level be greater than benefits for the comprehension it may provide? Is it not needed at all? We did not found conclusive evaluations answering these questions.

As we pointed out earlier, our intention is to find a balance between the cost required to explicitly preserve the programmer's mental model by annotating concerns and the value provided by those annotations. As a fist step, we previously conducted a case study [9] in which 5 participants (annotators) were annotating their own small-sized codebases with high-level concerns. We used custom annotating tool created for the Atom[2] code editor through which annotators assigned named and colored tags to arbitrary code fragments (see fig. 2). To briefly sum the study up, we found that:

- Code fragments were annotated at *expression*, *statement*, *method/field*, and *class* granularity levels.
- *Method/field* and (interestingly) *statement* levels were used by all participants, and represented the main body of the annotations.

While we do not consider the study to be conclusive, the results at least suggest that usefulness of the statement concern granularity level should be further evaluated. However, we will need to prepare appropriate tool for such evaluation, as no existing one allows us to do them at that granularity level, or to perform a fair comparison between multiple granularities due to workflow differences.

During the study we also observed some disadvantages of free-form annotation approach, where arbitrary code fragments can be annotated: annotations not making sense from the code structure point of view were also made. For example, multiple statements within a method were selected, but the first and the last statement were not selected entirely. Were the boundary statements intended to be selected whole? Or was some of them selected unwittingly by quick, but imprecise selection?

The free-form annotation strategy thus seems to require a great amount of attention that could be better spent elsewhere. And although the strategy provides the most flexibility, concern annotations for a given software system may benefit from some rules regarding which elements can be annotated. A more effective solution for experimentation with different annotation granularity levels could be to have code fragment selection automatically adjusted to match configured annotation rules.

IV. A Concept for Unified Code Annotation Tool

So far we discussed various aspects of code annotation approaches and showed that there is no single best annotation type or annotation granularity level. In this section we present a concept for an annotation tool that will support configurable set of annotable elements and multiple annotation types. Differences of individual annotation types will be abstracted by uniform annotation workflow and user interface (UI). This will allow us to evaluate the cost and benefits of preserving concerns at different granularity levels, while keeping the annotation approach the same.

We are implementing a prototype of the tool concept as a plugin for the *IntelliJ IDEA*[3] IDE.

A. Configurable Annotation Granularity Levels

Following the insight on free-form annotation issues gained during our above described study, the set of elements annotable by the tool will be restricted, but it will be configurable by code *structural patterns*. This will ensure that annotated code fragments will be structurally valid. It will also enable simpler evaluation of different granularities of annotable elements by supporting them with a single tool.

The structural patterns will be based on the structure of a special abstract syntax tree (AST) called *Program Structure Interface* (PSI).[4] The PSI is built by the *IntelliJ* platform for the edited code and is used for all code structure-aware features, like code completion and refactoring [11]. Alongside the PSI itself, the *IntelliJ* platform provides an application programming interface (API) in a form of an internal DSL (with Java as the host language) for tree pattern matching against the PSI. An example of a pattern in this DSL for matching method parameters can be seen in listing 3.

Listing 3. Elements of method parameters matched with *IntelliJ* PSI tree pattern-matching API

```
psiElement(PsiParameter.class).withParent(
  psiElement(PsiParameterList.class)
    .withParent(PsiMethod.class)
);
```

Possibilities of the tree pattern matching API go beyond what is required for specifying set of annotable elements. We will use simpler custom DSL to denote the patterns in a tool's configuration file and translate them to *IntelliJ* APIs with an interpreter. An example of full annotable patterns configuration for Java *statement*, *method* and *class* elements is shown in listing 4. There, `JAVA` is the Java language identifier on the *IntelliJ* platform and `com.intellij.psi` is the package where classes of its PSI elements are located. The package name is needed to obtain fully qualified class names for the interpreter. Syntax of the patterns themselves is inspired by the *XPath* language.

B. Uniform Annotation Workflow

Code annotation workflow consists of several steps: selecting a code fragment, annotating the fragment, and managing existing annotations during code evolution. In our tool we want to unify these steps across different annotation types in order to provide uniform annotation workflow.

[2] Available at https://atom.io.

[3] Available at https://www.jetbrains.com/idea.

[4] The PSI is a special AST in a sense that while it is more abstract than a parse tree, it preserves all parse tree tokens.

Listing 4. Example configuration of annotable elements for Java language

```
patterns for JAVA : com.intellij.psi {
  statement = PsiCodeBlock//PsiStatement;
  method = PsiMethod;
  class = PsiClass;
}
```

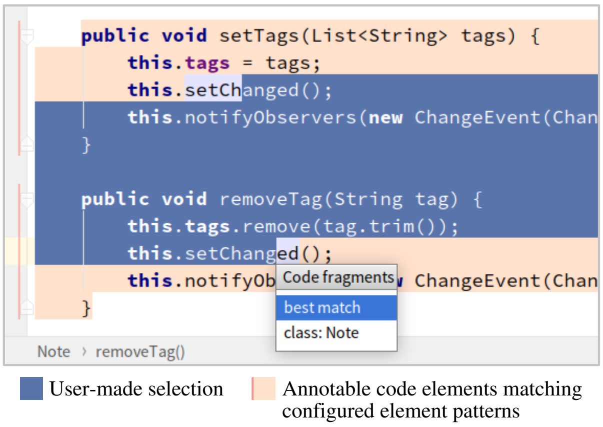
1) Selecting a code fragment: In order to unify the process of annotating code with different types of annotations, we can use only annotation methods not requiring direct code change: *selection* (e.g., like our tool for the earlier mentioned study [9]) or *drag & drop* actions (e.g., like *Concern Mapper* [8]). As we want to minimize the code annotation friction with the typical programmers' workflow, we will use the selection method because it can be performed with both a mouse and a keyboard. Still, two general cases may occur: we can annotate elements according to either an editor caret position (empty selection), or non-empty text selection.

The first case with an empty selection is simpler to resolve. PSI pattern matching is performed from the element under the caret with all configured patterns, and resulting matched elements are sorted by their depth in the element tree in descending order. The second case is more complex: an arbitrary user-made text selection needs to be resolved into a list of pattern-matched elements. For example, a selection can start in the body of one method and end in the body of the following method. Considering that patterns from listing 4 are in place, selection of whole methods, as can be seen in fig. 3, is the closest continual and valid pattern match.

The element resolving algorithm in case of a non-empty selection proceeds as follows:

1) Take all visible leaf elements in the selected text range.
2) Map each element into a list of elements matching some of configured patterns.
3) Flatten the result into a set of distinct elements.
4) Construct the final list of annotable fragments that has
    a) the first fragment consisting of such matched elements that are highest in the element tree and at the same time partially covered by the selection,
    b) following fragments created from each matched element that contains the whole selection, sorted descendingly by their depth in the element tree.

Whenever more than one annotable fragment is resolved, the user can select the desired one either by fragment extending/shrinking actions[5] (moving forward or backward within the list of resolved fragments, respectively), or by action where direct fragment selection is possible through a popup dialog (as can be seen in fig. 3). Such annotation process comes close to the *Commands composition pattern*, used in editors like *Vim*, as described by Chodarev [12]: an *action* (to select a code fragment) followed by an *object* (fragment matching one of the patterns) the action should be applied to.



```
    public void setTags(List<String> tags) {
        this.tags = tags;
        this.setChanged();
        this.notifyObservers(new ChangeEvent(Chan
    }

    public void removeTag(String tag) {
        this.tags.remove(tag.trim());
        this.setChanged();
        this.notifyOb                  w ChangeEvent(Chan
    }
```

🟦 User-made selection 🟧 Annotable code elements matching configured element patterns

Fig. 3. User-made code selection resolved into annotable elements, with current seection highlighted and other available selections shown in a popup

2) Annotating selected fragment: Annotating a code fragment means assigning it to a user-defined annotation. Such annotation has a name and an optional description. The annotations belonging to a particular code fragment can be displayed within the editor gutter, as were in our Atom editor extension (see fig. 2). If configured to do so, annotation can be also added to the code as either LLA or structured comment.

To mitigate the noise created with annotation comments at least on a view level, the *code folding* editor facility can be leveraged. There are examples of advanced custom code foldings used to make code more readable[6]. Similarly they will be used by our tool: they can automatically *fold* annotation comments, rendering them *not visible* in the code.

Code fragments annotated with external annotations can be stored as annotated text range offsets in a file or a database. But recovering from changes made without the tool will be too difficult without any additional anchor points to the annotated code. These can be, however, created from names either of the annotated elements directly or of their enclosing parents.

3) Managing annotations: Existing annotations will be managed through custom IDE views and actions, from where they will be available for reuse on other code fragments. The tool will also provide *refactoring* and *search* operations. Moreover, the tool should be able to assist in cases like moving already annotated code: selections of code elements with internally persisted annotations will be automatically adjusted to cover also the annotations.

C. Leveraging Concern Annotations in IDE Tools

Our motivation for having the high-level concern annotations available within an IDE is to leverage them for program comprehension. Eventually, our intention is to use them to drive dynamic code projection, e.g., similar to the one available in the SSCE tool [3] (more details on the projection are

[5] These actions are similar to *extend/shrink selection actions* available in *IntelliJ IDEA*, documented at https://www.jetbrains.com/help/idea/selecting-text-in-the-editor.html, but they select only elements matching one of configured patterns.

[6] See the plugin introduced at https://medium.com/@andrey_cheptsov/making-java-code-easier-to-read-without-changing-it-adeebd5c36de.

given in section V). But simpler projections that can augment existing IDE views are also possible.

IntelliJ platform provides a multitude of extension points (EP) that can alter data in various IDE views. For example, `TreeStructureProvider` and `StructureViewBuilder` EPs can be implemented to make project tree structure and file structure views aware of concerns present in individual files and filter them according to a concern selection. Furthermore, custom code foldings can hide all the code unrelated to the selected concerns. Altogether, the idea is to utilize concern annotations with appropriate EPs to make the IDE tools *concern-aware*.

V. Related Work

In this section we look at supported granularity levels and other features of existing tools related to our presented tool concept. We also discuss other works related to code annotations and to our tool concept in general.

Storey et al. [7] present their *TagSEA* tool for annotating code with special comments—tags. These tags are associated with the nearest enclosing code element, with fines granularity of class members. However, coarser *file* and *package* granularity levels are also supported, using external annotations. The tool is implemented as *Eclipse* IDE plugin and provides views for filtering the tags, for displaying all the locations of a particular tag, and for combining tags into personalized *routes*, providing thus user-specific navigation around the tagged code. For tag management, refactoring and auto-completion is supported. Although the tool supports both *structured comments* and *external* annotation types, they are managed differently; comment annotations are added as text, external annotations through context menus of project files and packages.

Robillard and Weigand-Warr's *Concern Mapper* [8] is a simple view-based *Eclipse* IDE plugin by which a programmer can register program elements with identified concerns. Supported elements are class fields and methods. While assigning concern to a class is not supported, a class is shown as belonging to a concern if all its members are associated with it. Concerns are created and managed through a separate tree view and associated with code elements by *drag & drop* actions. Similar to our idea of integrating the concerns with IDE tools, the *Concern Mapper* augments the *Search* and *Project* views of the hosting IDE.

Kersten and Murphy's *Mylyn* (formerly Mylar) [13] is task-based tool capturing programmers' activities within the *Eclipse* IDE into a *Degree of Interest* model. They use gathered data to augment IDE views (e.g., search or structure view) in a similar way to what we proposed to do with annotations.

Robillard and Murphy present *Feature Exploration and Analysis Tool* (FEAT) [10] for extracting abstracted program model from the code and representing concerns in that model. The selection of elements takes a specific approach: it is done though queries against the extracted program model and stored in concern model. The model supports the usual class and class members code elements. Views for accessing the program and concern models are available. We currently do not have any specific concern model prepared for our tool, except for defining a hierarchy of concerns.

Porubän and Nosál' [3] in their *Sieve Source Code Editor* (SSCE) tool provide LLA-based code projection built of all elements annotated with the chosen annotations into a single view—a code projection editor. The tool supports the Java language, with annotation granularity restricted to classes and class members. This editor is complemented with a view for managing concerns identified in the code and for selecting which concerns should the projection be built of. A significant disadvantage of the projection editor, caused by limitations of hosting *NetBeans* IDE, is that it loses many IDE features (e.g., code completion). We plan to explore possibilities of the *IntelliJ* platform for keeping such essential features functional for code projections.

Our concept of an annotation tool tries to eliminate disadvantages of individual annotation types by allowing to choose an appropriate type for a given annotation or use case. Work of Cazzola et al. [14] goes in the direction of modifying the Java compiler to allow the formal LLA on finer granularity levels than available with the standard compiler.

We propose to use AST patterns to ensure structural validity of annotated code fragments. Kästner et al. [15] and Behringer et al. [16] use AST rules to ensure syntactically correct separation of features in feature-oriented development of software product lines. Kästner et al. call the annotation process *coloring*, and Behringer et al. extend it with *snippet* code organization system for managing feature variability.

VI. Future Work Direction

The concept of a code annotation tool presented in this paper is a continuation of our work towards leveraging custom code metadata that capture high-level details from the problem domain in IDE tools. Augmentation of these tools can make the IDE concern-aware, assisting programmers during code comprehension on a higher level of abstraction.

We are implementing a prototype of the presented tool concept as *IntelliJ IDEA* IDE plugin. Element pattern interpreter and code selection facility were already implemented and tested on Java and Python programming languages. The next work on the prototype covers storing created annotations externally. With a working prototype of uniform annotation workflow we plan to evaluate its usability, reflect the results in the implementation, and gradually add concern-aware extensions to IDE tools as described in section IV-C. With the extensions in place, we can start to evaluate usefulness of different granularity levels configurations for program comprehension tasks, as a follow-up to our previous study.

A long-term goal is to leverage the concern annotations for implementation of a custom code view—a concern-oriented code projection [3]—to further facilitate the comprehension process.

Acknowledgment

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10.

REFERENCES

[1] W. Maalej, R. Tiarks, T. Roehm, and R. Koschke, "On the Comprehension of Program Comprehension," *ACM Transactions on Software Engineering and Methodology*, vol. 23, no. 4, pp. 31:1–31:37, aug 2014.

[2] M. Nosáľ, J. Porubän, and M. Nosáľ, "Concern-oriented source code projections," in *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, Kraków, 2013, pp. 1541–1544.

[3] J. Porubän and M. Nosáľ, "Leveraging Program Comprehension with Concern-oriented Source Code Projections," *3rd Symposium on Languages, Applications and Technologies. OpenAccess Series in Informatics (OASIcs)*, vol. 38, pp. 35–50, 2014.

[4] V. Vranić, J. Porubän, M. Bystrický, T. Frťala, I. Polášek, M. Nosáľ, and J. Lang, "Challenges in Preserving Intent Comprehensibility in Software," *Acta Polytechnica Hungarica*, vol. 12, no. 7, pp. 57–75, dec 2015.

[5] B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk, "Feature location in source code: a taxonomy and survey," *Journal of Software: Evolution and Process*, vol. 25, no. 1, pp. 53–95, jan 2013.

[6] M. Sulír, M. Nosáľ, and J. Porubän, "Recording concerns in source code using annotations," *Computer Languages, Systems and Structures*, vol. 46, pp. 44–65, nov 2016.

[7] M.-A. Storey, L.-T. Cheng, I. Bull, and P. Rigby, "Shared waypoints and social tagging to support collaboration in software development," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work - CSCW '06*. New York, New York, USA: ACM Press, 2006, pp. 195–198.

[8] M. P. Robillard and F. Weigand-Warr, "ConcernMapper: simple view-based separation of scattered concerns," in *Proceedings of the 2005 OOPSLA workshop on Eclipse technology eXchange - eclipse '05*. New York, New York, USA: ACM Press, oct 2005, pp. 65–69.

[9] J. Juhár and L. Vokorokos, "Separation of concerns and concern granularity in source code," in *2015 IEEE 13th International Scientific Conference on Informatics*. IEEE, nov 2015, pp. 139–144.

[10] M. P. Robillard and G. C. Murphy, "Representing concerns in source code," *ACM Transactions on Software Engineering and Methodology*, vol. 16, no. 1, p. 38, feb 2007.

[11] D. Jemerov, "Implementing refactorings in IntelliJ IDEA," in *Proceedings of the 2nd Workshop on Refactoring Tools - WRT '08*. ACM Press, oct 2008, pp. 1–2.

[12] S. Chodarev, "Commands composition user interface pattern," in *2015 IEEE 13th International Scientific Conference on Informatics*. IEEE, nov 2015, pp. 120–123.

[13] M. Kersten and G. C. Murphy, "Mylar: a degree-of-interest model for IDEs," in *Proceedings of the 4th international conference on Aspect-oriented software development - AOSD '05*. ACM Press, mar 2005, pp. 159–168.

[14] W. Cazzola and E. Vacchi, "@Java: Bringing a richer annotation model to Java," *Computer Languages, Systems and Structures*, vol. 40, no. 1, pp. 2–18, 2014.

[15] C. Kästner, S. Apel, and M. Kuhlemann, "Granularity in software product lines," in *Proceedings of the 13th international conference on Software engineering - ICSE '08*. New York, New York, USA: ACM Press, 2008, p. 311.

[16] B. Behringer, L. Kirsch, and S. Rothkugel, "Separating features using colored snippet graphs," in *Proceedings of the 6th International Workshop on Feature-Oriented Software Development - FOSD '14*. New York, New York, USA: ACM Press, 2014, pp. 9–16.

Proposal of Human Body Description Format XML Schema and Validation of Anthropometric Parameters

O. Kainz*, F. Jakab*, M. Michalko*
*Department of Computers and Informatics, Košice, Slovakia
ondrej.kainz@tuke.sk, frantisek.jakab@tuke.sk, miroslav.michalko@tuke.sk

Abstract — **In this paper, the development of XML schema of newly proposed Human body description format (HBDF) is proposed and further, the software solution for processing the anthropometric parameters of the human body is presented. The analysis covers information on the HBDF format and specific measurement procedures that are associated with the format and together form a coherent structure for the recording of such data. The structure of the format itself is implemented as XML schema, determining the content of XML outputs. Experimental software solution is designed and proves valid in the examination of HBDF structure. Schema ensures the data coherency and integrity, while the proposed solution ensures the import, operation, and export of data. The XML schema was experimentally tested as a part of multiple research projects.**

Keywords—anthropometric parameters; human body description format; ISO standard; xml schema.

I. Introduction

Effective and fast processing of measured data on parts of the human body is currently a major challenge of many scientific fields. Technological advances in this area are the only indicators of the recent period; e.g. 3D body scanning of the human body, monitoring systems of the human body, various consoles with Kinect sensors, and so on. All these approaches are implemented using the latest sensors and technological innovations, combined with advanced software systems and algorithms.

In addition to state-of-the-art approaches, traditional methods of measuring parts of the human body exist; these are the tailoring measurements or measurements of anthropometric parameters using various instruments and equipment, e.g., anthropometer.

Measurements of parts of the human body of this particular type are performed manually at precise points, thus the accuracy of it is an important factor. During the manual measurements, one of the biggest challenges is inscription, storing and displaying the data after performing various measurements. For this very particular reason, it is very important and appropriate to organize the data obtained from different sources and devices of different types into such a specific structure so that the measurement results can be further processed and presented based on certain requirements.

The main objective of this research is therefore to design and implement a system, which is to serve for the recording of anthropometric parameters of the human body and is able to put these data into precisely defined structures. It is necessary to study the adequate theory in connection with this issue and, consequently, to select technologies that are ideal for the implementation of the specified application requirements.

Another important goal of the research is not only to provide a software solution for the processing of measured data of the human body but also to provide solutions that might be in the future, linked with other researches dealing with the same topic.

II. Analysis of Human Body Measurements

The diagnostics or measurement of human body dimensions can be divided to traditional [1] and more modern, 3D body scanning [2]. Moreover, excessive attention to areas dealing with anthropometry [3], as part of biometric analysis, is equally important. All of these approaches are described in more detail in next parts.

A. The technique of measuring the dimensions of the human body in tailoring

An area where it is important to conduct the most accurate measurement possible, in order to achieve the most accurate results, is tailoring. Furthermore, concerning the dimensions in trimming, it is also necessary to calculate different aspects, such as pattern.

Basic dimensions include length, width and circumference dimensions, where length dimensions are inscribed in full merit, while the other two are round off by the first decimal places [1]. Due to possibility of asymmetry-likeness of the human body, it is advisable to conduct the measurement on both sides simultaneously.

B. The technique of measuring the dimensions of the human body in the application of anthropometry

Measurement of height, length and width requires a thorough knowledge of the anatomy of the human body [4]. There are fifteen anthropometric points [5] on the human body (see Fig. 1), which are detected by applying the projection, i.e. based on precise anthropometric points from the base, i.e. where the individual is standing or sitting. These points are specified also in [5] and [6].

III. Mobile Application Design

The primary task of the mobile solution is to ensure that the measured data of the individual are processed. The goal is to implement and realize a feature such as inscription, viewing, updating and deleting records from the database. Consequently, questions on generating the output, using a

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

mobile application are formulated. For this purpose, an extensible XML mark-up language was chosen, due to its well-defined structure, where the XML output structure itself is implemented, using the XSD XML schema [7].

The software design consists of three parts. Each individual part has its own specific function, which is as follows: mobile application user interference (handling measured input values); database design (storing measured data in the database); and the processing of measuring input data (e.g. based on anthropometric points located on the human body) organized into unique XML structure.

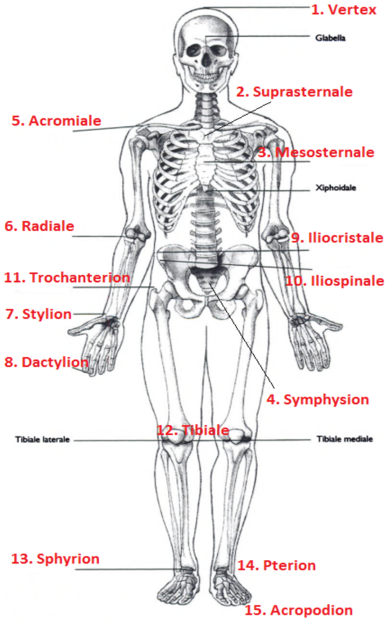


Fig. 1. Anthropometric points of the human body [6].

A. *Mobile application interface*

The main task of the mobile application is to present a list of processed data, in order to ensure data manipulation in the form of – Insert, Update, Delete – and to secure the export of output data in structured XML formats. Based on the above-mentioned facts, in-app user windows can be divided into multiple groups, based on functionality requirements, which are as follows:

1) Forms to inscript records to databases: these are secured with database functions, including Insert for inscribing a record.

2) Forms to display records from databases and to edit them in the database: these include functions ensuring the presentation and activation of records using activities and fragments. Implementing the presentation of records stored in databases is conducted through:

- Registry: *DevListFragment*, *InfoListFragment* and *MeasListFragment*.
- Graphs in the form of Barcharts: a graph presenting individual measured parts of the human body; a graph presenting the number of measured data found on different parts of the human body (relating to a specific individual); a graph presenting specific measured parts of the human body (concerned with each and every individual); and a graph presenting the number of measured data on different parts of the human body (similarly including each
- Export and validation window - methods for exporting entries such as: methods for creating and XML structure, inscribing records to a file, and saving a file to a mobile device´s memory card. Validation method with the parameters of the uploaded XML file and XML schema; an XML output check is being conducted, relative to the XML schema itself.

B. *Database*

Since the mobile solution enables the manipulation of measured data, it is adequate to organize the data and results into individual tables. To achieve this, a database was designed to test the mobile solution and store the information with the help of a mobile application. First, it was necessary to choose a suitable database that meets all the above-mentioned requirements. For this reason, a SQLite database is relatively easy to use and is part of the Android operating system. It is a minor but at the same time powerful database capable of storing data on the user´s local server.

The HBDF database (Human Body Description Format) consists of three tables: TypeOfDevice, PersonalInformation and TypeOfMeasurement. Each table consists of a variety of information characterizing the given table, such as device information (ID, device names); information of the individual (ID, name, weight, height, date of measurement, type of device); measurement information (ID, type, part, measurement, value, individual). The grouping of these tables serves, in addition to preserving various types of information, to fill up the structure with information of the XML document itself.

C. *XML structure and the Schema*

The hierarchical XML model is enabled by using the HBDF schema as proposed in [8], see Fig. 2, defining the structure of the XML outputs, including various measurements conducted on the human body. The XML schema is used to define the XML structure; therefore it is necessary to design a custom-like XML schema as precisely as possible, which in later stages serves to validate various XML inputs.

XML documents have a form of a tree structure that starts at "the root" and branches to a smaller "leaves". From the scheme, it follows that the structure is enabled through the complex and non-complex elements. From this it is obvious that HBDF is a complexType as well as a root element of this particular scheme, consisting of so-called sub-elements (also complexType), where the sequence element defines the exact order of the *TypeOfDevice*, *PesonalInformation* and *TypeOfMeasurement*. The previously mentioned three sub-elements forms the structure of XML documents itself, and thereafter this structure is filled up with data from the HBDF database.

```
<xs:element name="HBDF">
 <xs:complexType>
   <xs:sequence>
    <xs:element ref="TypeOfDevice"/>
     <xs:element      ref="PersonalInformation"/>
     <xs:element ref="TypeOfMeasurement"/>
   </xs:sequence>
 </xs:complexType>
</xs:element>
```

These elements are subsequently subdivided into smaller subdivisions and "leaves" on the basis that the information is retained. In the case of the *TypeOfDevice* component, parcelling is to define a complex device type that consists of two simple elements, including ID and device name. The occurrence of the Device element is set to minOccurs = "1" and maxOccurs = "unbounded" which means that the minimum number of repetitions equals one and the maximum occurrence is unlimited. It is important to note that in case of a simple element, this element similarly represents “leaves” of the tree structure, since such element has no descendent, but only the value holding the element itself. The following section of the XML schema describes the above-mentioned simple ID elements of Integer-type and name of String-type, where the rule applies that each item (except ID) is mandatory.

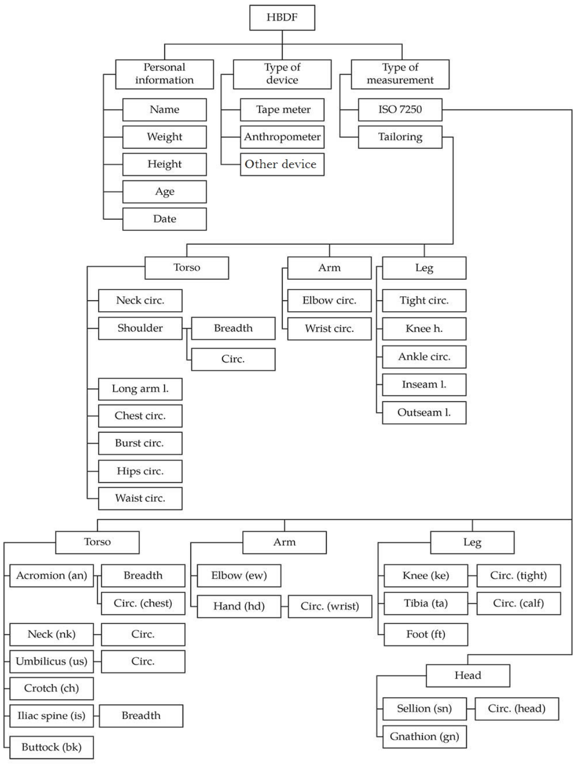


Fig. 2. HBDF scheme [8].

In the resolution of the element *PersonalInformation*, the same procedure is used, however with the difference that in this particular case for a complex type of person, consisting of six simple elements, a device attribute has been added, serving as a reference to a specific device. From the scheme, it follows that in the resulting structure, the occurrence of the device Attribute is not necessary, since its usage is optional. In the definition of the *min* and *maxOccurs* the procedure is identical to the *TypeOfDevice* element, which also applies to other later stages in the structure, where these functions are being used. The bellow presented part of the XML schema depicts the sub-elements for the main element *PersonalInformation*.

```
<xs:element name="person">
 <xs:complexType>
  <xs:sequence>
   <xs:element minOccurs="0" maxOccurs="1"
     ref="id"/>
   <xs:element ref="name"/>
   <xs:element ref="weight"/>
   <xs:element ref="height"/>
   <xs:element ref="age"/>
   <xs:element ref="date"/>
  </xs:sequence>
  <xs:attribute name="device" use="optional"/>
 </xs:complexType>
```

The creation of a structure for *TypeOfMeasurement* is irregular in that sense (as is already shown in *Fig. 2*) that in this case it is a much more complex structure. The complex element *TypeOfMeasurement* continues to consist of several complex elements, which are then processed into simpler elements (no longer complex rows). Thus, it is appropriate to classify specific complex elements into multiple levels and work oneself towards a specific hierarchical tree structure. The root of the tree is formed by the above-mentioned *HBDF*, in which the *TypeOfDevice* and *PersonalInformation* nodes are independently separated, with the *TypeOfMeasurement* node (the root of the sub tree) and consists of various descendants on the basis what measurement took place and on what particular part of the human body. In the case of the *Tailoring* method, three descendants are being recognized: *Torso*, *Arm* and *Leg*. However, the same structure and the same rules apply to *ISO_7250*, with the difference of defining the part *Head*. Each of these parts is then composed of different types of scale, which represent the next level of this structure, being an equally complex type. Furthermore, this node consists of a measurement element, which, in this case, represents the String type name; from the simple element value, which is the decimal value; and from the person attribute, that represents the reference to a particular person from the *PersonalInformation* element.

Element measurement can be both simple and complex. If it is a simple type, the element measurement ends on this particular level (it does not specify any descendent). In rare cases see Fig. 2, measurements may be divided into other parts, or types of measurements. An example is the *Tailoring* method, where the *Torso* (a particular part of the human body - *Shoulder* type) can be divided into *Breadth* and *Circ*. In similar cases like the previous one the structure Shoulder becomes a type of measurement attribute, and the measurement element becomes a specific part for the Shoulder type. The part listed below defines the structure definition from the scheme, in case the measurement is complex.

```
<xs:element name="measurement">
 <xs:complexType mixed="true">
 <xs:attribute name="type"/>
 </xs:complexType>
</xs:element>
```

IV. Implementation of the Proposed Software Solution

In order to meet the objectives of the proposed system, another necessary thing is the creation of a software solution in the form of a mobile application serving to describe the measured parts of the human body. A suitable application development tool is the chosen Java programming language, in which the main elements are the classes themselves. Given classes are of different types and can be divided within applications, by virtue of the function, as follows:

A. Class Activities

Their role in Android is to meet the goal of creating a presentation layer within applications. The following fact applies: one activity represents one window, which is mostly realized on the full screen. Each activity in its source code inherits its attributes from the parent class Activity, from the package *android.app*.

1) MainActivity: it is displayed right after the strat of the application and serves for the restortation of all the measurements in the database.

2) FileexplorerActivity: it is displayed when the Browser button is pressed and provides work with files that are placed in the internal memory of the mobile device.

3) ChartActivity: it is displayed when the Graphs button is pressed and provides the presentation of measured data using barcharts. The class itself extends ListActivity to view the list of graphs, of which four different types are implemented in the app. The chart interference itself is secured with achartengine.

B. Class Fragments

Fragments indicate a new way of creating user interference, located between activity layers and views. The parent class Fragment is located within *android.app.Fragment*, where the very behaviour of the fragment itself is being implemented. Fragments represent such independent Android components that can be used with the mentioned activities. Additionally, an activity can also host live fragments at a time. The fragment itself is in the context of the individual activities and its life cycle is linked to the life of the activity in which the fragment is located.

1) DevAddFragment is for the inscription of device name, InfoAddFragment for the iunscription of information about individuals, and MeasAddFragment for the inscription of measurement data: they consist of different spinners, formatters (for data) and buttons for adding records to the database. After a successful record saving into the database a message appears indicating message saving via a Toast message.

2) DevListFragment is for the displaying of a list of devices, InfoListFragment to display the liost of individuals, and MeasListFragment to display the list of measurement: the very views themselves are presented using the ListView list, where, on the basis of queries, methods such as the `onItemLongClick()`, which launches the delete operation and `onItemClick()`, which , while clicking onto the record itself, opens a dialogue to update this record.

3) InfoDialogFragment and MeasDialogFragment: are special Fragments that are used when it is necessary to display a contextual modal window within an activity or fragment. This solution ensures updating of the database record.

C. Database classes

The *android.database* package is specified to work with the Android database itself. In order to meet this goal, a DataBaseHelper was developed, expanding the SQLiteOpenHelper class, which provides database management. The purpose of using the database is to store data: device name, personal data of the individual, and information about the measured data, where the class implements the `onCreate(SQLiteDatabase)` method used to create a new database – that is when the application starts for the first time, and it is necessary to create tables with specific columns.

D. DAO and Entity classes

The MeasurementDBDAO class is used to retrieve a local database using the `getWritableDatabase()` method over the DataBaseHelper class, in order to achieve goals such as inscribing into the database and reading from it by using the SQLiteDatabase object that represents the database itself. Subsequently, below presented DAO classes are being displayed, extending the MeasurementDBDAO class while using the SQLIteDatabase in order to secure the SQLite CRUD operation, such as Create, Read, Update and Delete.

1)DeviceDAO: the primary task of this class is to store the device record in the database and to obtain a list of devices; subsequently, this list is called when entering personal data.

2)InfoDAO: this class, in addition to the aforementioned operations and functions, also performs the updating and deletion of personal data. In addition, it returns a list of personal data that is used when inserting measurements into the database and updating measurements.

3) TypeDAO: this class, like the previous one, provides for storing and updating measurement information into the database, as well as obtaining the list of measured data for the application on the main screen where the user is able to view all the stored data in the database.

Entities in the application are used to create objects to manipulate data while also playing an important role in working with the database.

1)TypeOfDevice: this class represents the TypeOfDevice table in the database and defines necessary device entries (integer ID and device name).

2)PersonalInfomartion: the database is represented by the PersonalInformation table and includes all the necessary items for the processing of personal data (integer ID, name, weight, height, age, date of measurement, and TypeOfDevice object).

3) TypeOfMeasurement: the database represents the TypeOfMeasurement table and specifies the data elements important for the processing of measurememnts (integer ID, type of measurement, part of measurement, name and value of the measurement itrself, and a PersonalInformation object).

V. Testing of the Implemented Solution

Solution evaluation specifies the testing and functionality of application implementation and the validation of the XML outputs with respect to the XML schema. Data generation is conducted through utilization of an application, while data itself can be obtained from various sources of equipment where measurements are carried out using various methods. Measurements are performed on different parts of the human body – on the head, on the torso, on the hands and on the legs.

The application is able to generate entered data into the system having a predefined structure, while being able to validate XML files generated by other sources or applications from other users.

Once the validation is running, the process may end either successfully (it is a valid XML file) or as an error message (unsuccessful), displaying the reason for not accepting an XML file owing to the XML schema.

A. *Export of measurements*

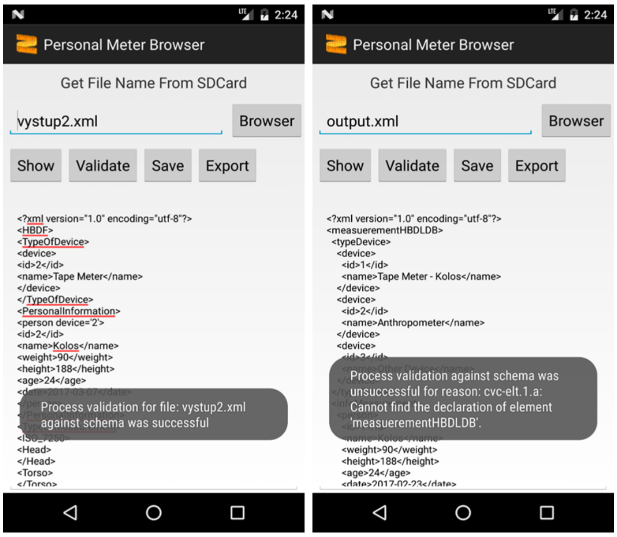
The XML output below indicates that the structure consists of three different segments: *TypeOfDevice* (defines the measuring device itself); *PersonalInformation* (depicts the person´s personal details, subsequently assigned to measurements on the human body); and *TypeOfMeasurement* (containing measurements conducted on the human body; in this case, the measurement is carried out on the troop). This output is generated through the application and represents the output of one particular individual, divided as follows:

```
<?xml version="1.0" encoding="utf-8"?>
  <HBDF>
    <TypeOfDevice>
     <device>
      <id>1</id>
      <name>Tape Meter</name>
     </device>
    </TypeOfDevice>
    <PersonalInformation>
     <person device='1'>
      <id>1</id>
      <name>Kainz</name>
      <weight>90</weight>
      <height>193</height>
      <age>24</age>
      <date>2017-09-12</date>
     </person>
    </PersonalInformation>
    <TypeOfMeasurement>
     <Tailoring>
      <Torso>
       <extent person='1'>
        <measurement type='Shoulder'
          Breadth(sh)
        </measurement>
        <value>100</value>
       </extent>
      </Torso>
     </Tailoring>
    </TypeOfMeasurement>
  </HBDF>
```

B. *Validation of XML*

Validation of an XML file, with respect to the XML scheme, is secured using the `validateXMLSchema()` Exception is processed by calling this method in the form of a report of a error message to the user´s mobile device screen. The parameter methods comprise of two files stored in the internal memory of the mobile device – XML schema and XML file. First, within the method an instance for Schemafactory compiler is created, which serves to read the external schema representation and prepares the validation scheme. Secondly, instances are created, which serve for the reading of an XML file together with StreamSource schemes. In this case, StreamSource is a good choice, since in the XML special encoding characters can be expected and Streamreader could not handle such characters. Following this, a Schema object is being created, representing a series of limitations that can be checked against the XML documents. The specific validation process begins on the fifth line of the method where a Validator object is created, which checks the XML document owing to the scheme. Before the checking begins, a newValidator is called in order to validate all the limits set by the XML schema over the schema object.

If the validation process is successful, the user will see information that validation has taken place successfully (see Fig. 3, left). On the other hand, if validation (for some reason) was unsuccessful, the user will be displayed with an error information on the screen as an output of validation in the form of Exception (see Fig. 3, right).



```
<?xml version="1.0" encoding="utf-8"?>
<HBDF>
<TypeOfDevice>
<device>
<id>2</id>
<name>Tape Meter</name>
</device>
</TypeOfDevice>
<PersonalInformation>
<person device='2'>
<id>2</id>
<name>Kolos</name>
<weight>90</weight>
<height>188</height>
<age>24</age>
<date>2017-03-07</date>
<ISO_7250>
<Head>
</Head>
<Torso>
</Torso>
```

Process validation for file: vystup2.xml against schema was successful

```
<?xml version="1.0" encoding="utf-8"?>
<measuerementHBDLDB>
 <typeDevice>
  <device>
   <id>1</id>
   <name>Tape Meter - Kolos</name>
  </device>
  <device>
   <id>2</id>
   <name>Anthropometer</name>
  </device>
  <device>
   <id>3</id>
   <name>Other Device</name>
   <weight>90</weight>
   <height>188</height>
   <age>24</age>
   <date>2017-02-23</date>
```

Process validation against schema was unsuccessful for reason: cvc-elt.1.a: Cannot find the declaration of element 'measuerementHBDLDB'.

Fig. 3. Validation of the XML file.

Conclusion

In this paper, the system for storing and processing of human body parameters utilizing new HBDF format was presented. The paper presents proposal of experimental software solution implemented on the specific mobile platform. This application is designed to record anthropometric parameters of the human body and ensures the manipulation of the measured data inserted into the database. The specific relational database, which is part of the Android operating system, has been selected to meet this goal. The principal functionality of the application is to export stored data based on the above-mentioned XML scheme and XML file with respect to this scheme.

The validation request is important from the point of view of the users of the application, in order to check other XML outputs of different users and sources. For the sake of making the experiments effective, quickly and as transparent as possible, it is essential that all the above-mentioned outputs meet an identical standard. It has been proven that the proposed application meets all the necessary requirements. The created experimental mobile solution is designed for a specific area to explore and measure parts of the human body. It is possible to use it anywhere in real-world scenarios.

In the future research, we plan to expand the application and extent the graphical user interference, i.e. introduce more advanced presentation of data using graphs in the application itself.

ACKNOWLEDGMENT

This publication is the result of the Project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, Phase II., ITMS: 313011D232, supported by the Research & Inovation Operational Programme funded by the ERDF. We support research activities in Slovakia/This project is being co-financed by the European Union.

REFERENCES

[1] ŠČAVNICKÝ, P.: Študijné texty pre učebný odbor výroba konfekcie: Stredná odborná škola Lipany, 2014.

[2] TESS: 3D scanning: the future of beskpoke suit tailoring, 2016.

[3] HRSTKOVÁ, L.: Vyhodnocní Somatotypu Českých Reprezentantek V Alpských Disciplinách. Diplomová práca. Masarykova Univerzita v Brne: Fakulta športovných študií, 2010, pp. 29-49.

[4] Carter J.E.L. Part 1: The Heath-Carter Anthropometric Somatotype: Instruction Manual, 2002.

[5] CHRTKOVÁ, Š.: Optimalizace Antropometrických Vyšetření Ve Sportovní Laboratořitul. Technická univerzita v Liberci: Fakulta Přirodovědne-Humanitní a Pedagogická, 2011, pp. 29-40.

[6] NOVOTNÝ, J.: Sportovní antropologie a antropometrie, 2009.

[7] FLYNN, P.: The XML FAQ. 2016, pp. 54-77.

[8] KAINZ, Ondrej. 2017.: Pokročilé vizuálne a nevizuálne prístupy k určeniu parametrov viacrozmerných objektov z reálnej scény : doktorandská dizertačná práca. Košice : TUKE. p. 117.

Solving Mean-Shift Clustering Using MapReduce Hadoop

Maksat N. Kalimoldayev
Institute of Information
and Computational Technologies
Almaty, Kazakhstan
mnk@ipic.kz

Vladimir Siladi
Faculty of Natural Sciences
Matej Bel University
Banska Bystrica, Slovakia
vladimir.siladi@umb.sk

Maksat N. Satymbekov
Institute of Information
and Computational Technologies
Almaty, Kazakhstan
m.n.satymbekov@gmail.com

Lyazat Naizabayeva
Institute of Information and Computational Technologies
Almaty, Kazakhstan
naizabayeva@gmail.com

***Abstract*—Paper presents results of a practical experiment that was conducted in order to pursuit main objective - design and implement novel iterative MapReduce framework based on Hadoop technology. To study the effects of implementation parallel scientific applications, we deployed. Despite the fact that they did not show better performance than the same algorithm implemented in MPI, we conducted experiments to solve problems iterative computations using the Hadoop architecture. As a result of the experiments, the fail-safe behavior of Hadoop technology was revealed in the solutions of complex hadacies.**

***Index Terms*—Hadoop, MapReduce, Mean-Shift, MPI, HDFS**

I. Introduction

Paper presents results of a practical experiment that was conducted in order to pursuit main objective - design and implement novel iterative MapReduce framework [1] based on Hadoop technology [2], [3]. To study the effects of implementation parallel scientific applications, we deployed algorithm of mean shift clustering on Hadoop platform. The reminder of the article is structured as follows. The first section presents introduction. In second section we present matematical model mean shift clustering. In third section we describe Hadoop, MapReduce and HDFS. In fourth section we comprised parallel algorithm for solving the mean shift clustering. In fifth section "Implementation and evaluation" presents MapReduce implementations performance. Finally sections discussion and conclusions. Our work is based upon modifying the programming model and using Hadoop standard features. Cluster is a collection of data members having similar characteristics. The process of establishing a relation or deriving information from raw data by performing some operations on the data set like clustering is known as data mining. Data collected in practical scenarios is more often than not completely random and unstructured. Hence, there is always a need for analysis of unstructured data sets to derive meaningful information. This is where unsupervised algorithms come in to picture to process unstructured or even semi structured data sets by resultant. Mean Shift Clustering is one such technique used to provide a structure to unstructured data so that valuable information can be extracted. This paper discusses the implementation of the Mean Shift Clustering Algorithm over a distributed environment using ApacheTM Hadoop. Research result is the comparison of the result with different cluster settings and MPI model.

II. Mean shift clustering

The mean shift algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. Given n data points $x_i, i = 1, 2, ..., n,$ on a d-dimensional space R_d, the multivariate kernel density estimate obtained with kernel $K(x)$ and window radius h is

$$f(x) = \frac{1}{nh^d} + K(\frac{x - x_i}{h}) \qquad (1)$$

For radially symmetric kernels, it suffices to define the profile of the kernel $k(x)$ satisfying

$$K(x) = c_{k,d}k(||x||^2) \qquad (2)$$

Where $c_{k,d}$ is a normalization constant which assures $K(x)$ integrates to 1. The modes of the density function are located at the zeros of the gradient function $\nabla f(x) = 0$.

The gradient of the density estimator (1) is

$$\nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1} (x_i x) g(||\frac{x - x_i}{h}||^2) \qquad (3)$$

where $g(s) = -k^1(s)$. The first term is proportional to the density estimate at x computed with kernel $G(x) = c_{g,d}g([||x||^2])$ and the second term

$$m_h(x) = \frac{\sum\limits_{i-1}^{n} x_i g(||\frac{x-x_i}{h}||^2)}{\sum\limits_{i-1}^{n} g(||\frac{x-x_i}{h}||^2)} - x \qquad (4)$$

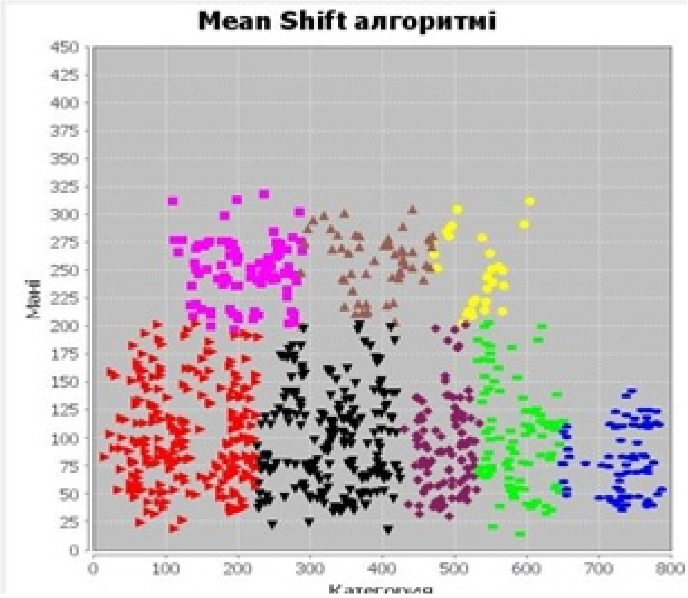


Fig. 1. Mean Shift.

is the mean shift. The mean shift vector always points toward the direction of the maximum increase in the density. The mean shift procedure, obtained by successive

- computation of the mean shift vector $m_h(x^t)$
- translation of the window $x^{t+1} = x^t + m_h(x^t)$

is guaranteed to converge to a point where the gradient of density function is zero. Mean shift mode finding process is illustrated in Figure 1. The mean shift clustering algorithm is a practical application of the mode finding procedure:

III. Hadoop, MapReduce and HDFS: A Developers Perspective

Hadoop is an open-source project overseen by the Apache Software Foundation Hadoop is mostly used for reliable, scalable and distributed computing, but can also be used as a general purpose file store capable of hosting petabytes of data. Many companies use Hadoop for both research and production purposes. Initially, Hadoop was primarily a tool for storing data and launching MapReduce tasks, but now Hadoop is a large stack of technologies that are somehow related to processing large data (not just with MapReduce). The core components of Hadoop are:

-Hadoop Distributed File System (HDFS) is a distributed file system that allows you to store information of almost unlimited volume.

- Hadoop YARN - a framework for managing cluster resources and task management, including the MapReduce framework.

Many of the features and configurations supported by Hadoop provide usability, power and wide-ranging application possibilities for this platform. For example, Yahoo! and countless other organizations are using Hadoop to analyze tons of bits and bytes of information. On the other hand, Hadoop can be easily launched on one node. All you need for this is the ability to program in Java (including generics).

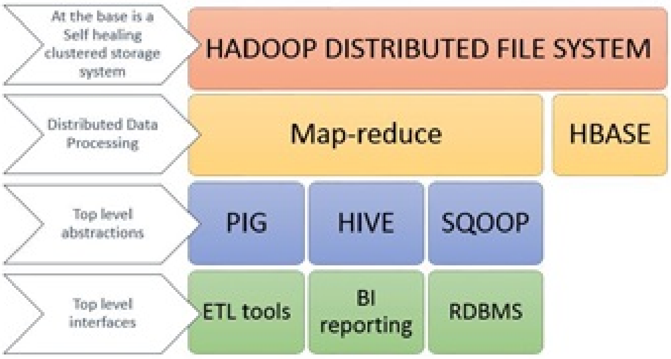

At the base is a Self healing clustered storage system
HADOOP DISTRIBUTED FILE SYSTEM
Distributed Data Processing
Map-reduce
HBASE
Top level abstractions
PIG
HIVE
SQOOP
Top level interfaces
ETL tools
BI reporting
RDBMS

Fig. 2. Hadoop Architecture.

In the Hadoop infrastructure, the map and reduce functions are implemented as an extension of the base classes. Specific implementations of functions are linked into a single module in accordance with the selected configuration, which, in addition, defines the input and output formats. Hadoop is well suited for processing huge files containing structured data. One of the main advantages of this platform is that it organizes the initial parsing of the input file, so you just have to process each line. Thus, the task of the map function is to determine which data is to be extracted from the incoming text line.

Hadoop handles large number of data from different system like Images, videos, Audios, Folders, Files, Software, Sensor Records, Communication data, Structured Query, unstructured data, Email conversations, and anything which we can't think in any format[5]. All these resources can be stored in a Hadoop cluster without any schema representation instead of collecting from different systems. There are many components involved in Hadoop like Avro, Chukwa, Flume, HBase, Hive, Lucene, Oozie, Pig, Sqoop and Zookeeper. The Hadoop Package also provides Documentation, source code, location awareness, Work scheduling. A Hadoop cluster contains one Master node and Many Slave nodes. The master node consists of Data node, Name node, Job Tracker and Task Tracker where slave node acts as both a Task Tracker and Data node which holds compute only and data only worker node. The Job Tracker manages the job scheduling. Basically Hadoop consists of two Parts. They are Hadoop Distributed File system (HDFS) and Map Reduce[5].HDFS provides Storage of data and Map Reduce provides Analysis of data in clustered environment. The Architecture of Hadoop is represented in figure 2

HDFS is the primary storage system used by Hadoop applications. HDFS creates multiple replicas of data blocks and distributes them on compute sites throughout a cluster to enable reliable, extremely rapid computations:

- Data is distributed across many machines at load time.
- HDFS is optimized for large, streaming reads of files rather than for irregular random reads.

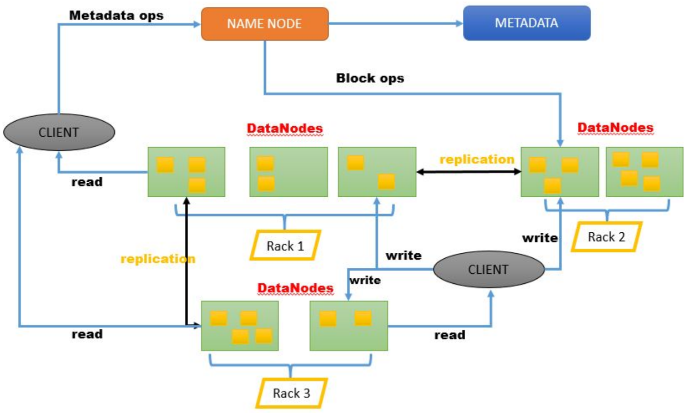


Fig. 3. HDFS Architecture.

- Files in HDFS are written once and no random writes to files are allowed.
- Applications can read and write HDFS files directly via the Java API.

The communication between the nodes occurs through Remote Procedure calls. Name node stores metadata like the name, replicas, file attributes, locations of each block address and the fast lookup of metadata is stored in Random Access Memory by Metadata. It also reduces the data loss and prevents corruption of the file system. Name node only monitors the number of blocks in data node and if any block lost or failed in the replica of a datanode, the name node creates another replica of the same block. Each block in the data node is maintained with timestamp to identify the current status. If any failure occurs in the node, it need not be repair immediately it can be repaired periodically. The architecture of HDFS is shown in Figure 3. Security Issues in HDFS The HDFS is the base layer of Hadoop Architecture contains different classifications of data and it is more sensitive to security issues.

MapReduce is a programming model and a framework for writing applications of large-scale computations on the basis of a large number of computations:

- Provides automatic parallelization and distribution of tasks.
- has built-in mechanisms for maintaining stability and operability in case of failure of individual elements.
- Gives status and monitoring tools.
- Provides a clean abstraction layer for programmers.

As an infrastructure designed to handle large data sets, the MapReduce model is optimized for distributed work across multiple computers. As the name implies, this infrastructure includes two main functions. The task of the map function is to split the incoming data stream into smaller sets and transfer these data sets to other processes for further processing. The reduce function analyzes the results of processing individual sets, obtained by the map function, and passes them to the output of the program.

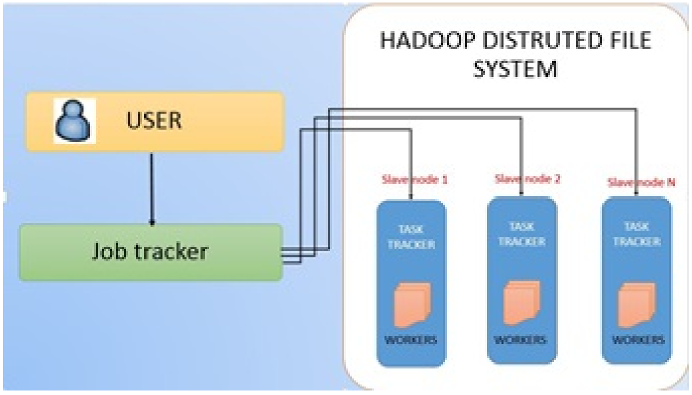


Fig. 4. Mapreduce Architecture.

The programmer designs a Map function that uses a (key,value) pair for computation. The Map function results in the creation of another set of data in form of (key,value) pair which is known as the intermediate data set. The programmer also designs a Reduce function that combines value elements of the (key,value) paired intermediate data set having the same intermediate key. [5] Map and Reduce steps are separate and distinct and complete freedom is given to the programmer to design them. Each of the Map and Reduce steps are performed in parallel on pairs of (key,value) data members. Thereby the program is segmented into two distinct and well defined stages namely Map and Reduce. The Map stage involves execution of a function on a given data set in the form of (key,value) and generates the intermediate data set. The generated intermediate data set is then organized for the implementation of the Reduce operation. Data transfer takes place between the Map and Reduce functions. The Reduce function compiles all the data sets bearing the particular key and this process is repeated for all the various key values. The final out put produced by the Reduce call is also a dataset of (key,value) pairs. An important thing to note is that the execution of the Reduce function is possible only after the Mapping process is complete. Each MapReduce Framework has a solo Job Tracker and multiple task trackers. Each node connected to the network has the right to behave as a slave Task Tracker. The issues like division of data to various nodes , task scheduling, node failures, task failure management, communication of nodes, monitoring the task progress is all taken care by the master node. The data used as input and output data is stored in the file-system.

IV. Parallel algorithm for solving the Mean-Shift Clustering

Algorithm 1 Mapper design for Mean-Shift Clusreing

```
procedure MeanShiftDesign
Load Cluster file
fp=Mapcluster file
 Create two list
 listnew=listold
CALL read(Mapcluster file)
newfp = MapCluster()
```

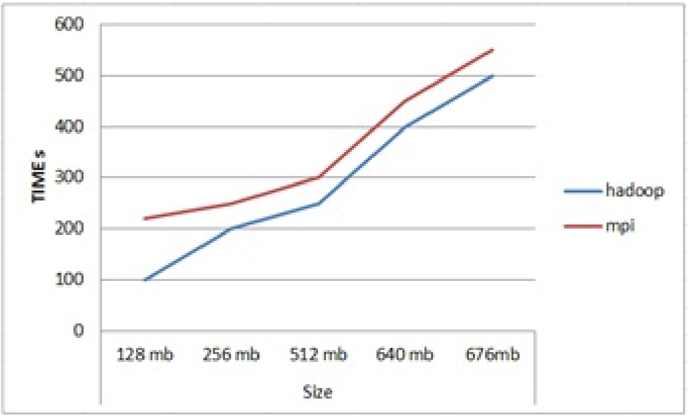


Fig. 5. Mapreduce Architecture.

```
dv = 0
Assign correct centeroid
  read(div)
calculate centeroid
div = minCenter()
  CALL MeanSHIFTReduce()
end procedure = 0
```

Algorithm 2 Reducer design for Mean Shift Clustering

```
procedure MeanShiftReduceClustering
NEW listofclusters
COMBINE result clusters from MAPCLASS.
if cluster size too high or too low then
RESIZE the cluster
$C_{Max}$ = findMaxSize(ListofClusters)
$C_{MIN}$ = findMinSize(ListofClusters)
if $C_{Max} > \frac{1}{20}$ totalSize then Resize(Cluster)
WRITE cluster FILE to output DIRECTORY
```

Algorithm 3 Implementing MeanShift Function

```
Procedure MeanShift Function
If Initial Iteration then LOAD cluster file from DORECTORY
else Read cluster file from previous iteration
Create new JOB
SET MAPPER to map class Defined
SET REDUCER to reduce class define
Paths for output directory
SUBMIT JOB
```

V. Implementation and Evaluation

In order to test the application we set up the infrastructure that included 8 Core i-5 Processor PCs with 16Gb RAM, HP Blade servers with 4 Core Intel Xeon Processors and Gigabit network connection Switch. All computing machines got 64-bit Ubuntu 12.10 operating system and Hadoop 1.2.1 platform installed.

VI. CONCLUSION

In this paper we studied the adaptation of scientific computing tasks to a cloud environment, such as MapReduce. The presented study presents a new iterative processing platform for Hadoop. Despite the fact that they did not show better performance than the same algorithm implemented in MPI, we conducted experiments to solve problems iterative computations using the Hadoop architecture. As a result of the experiments, the High-Availability behaviour of Hadoop technology was revealed in the solutions of complex problem.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[2] C. Lam, Hadoop in action. Manning Publications Co., 2010.

[3] T. White, Hadoop: The Definitive Guide. O'Reilly Press, 2009.

[4] Lammel, R.: Google's MapReduce Programming Model - Revisited. Science of Computer Programming 70, 1–30 (2008).

[5] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu,P. Wyckoff, R. Murthy, "Hive – A Warehousing Solution Over a Map-Reduce Framework," In Proc. of Very Large Data Bases, vol. 2 no. 2,August 2009, pp. 1626-1629

[6] M. Fashing and C. Tomasi. Mean shift is a bound optimization. IEEE Trans. Pattern Analysis and Machine Intelligence, 27(3):471–474, Mar. 2005.

[7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence, 24(5):603–619, May 2002.

[8] Apache Hadoop. http://hadoop.apache.org/

[9] Y. Wong. Clustering data by melting. Neural Computation, 5(1):89–104, Jan. 1993

[10] X. Yuan, B.-G. Hu, and R. He. Agglomerative mean-shift clustering. IEEE Trans. Knowledge and Data Engineering, 24(2):209–219, Feb. 2010.

aaltOS for Energy Harvesting Applications:

Effects of Clock Frequency and System Tick on Power and Energy Consumption

Ammar Ahmed Khan, Mika Ahistus, Tero Liukko, Joni Lumela, Otto Sassi, and Seppo J. Ovaska
Aalto University
Department of Electrical Engineering and Automation
Espoo, Finland
ammar.khan@aalto.fi, seppo.ovaska@aalto.fi

Abstract—**Real-time operating system (RTOS) is a common part of today's embedded systems. The use of RTOS provides an organized and maintainable programming environment for developers. In battery-operated wireless sensor nodes, the energy and power consumption measures have become of great importance. At the same time, the fundamental importance of real-time performance cannot be compromised. In this paper, we present a case study that clarifies the relation between energy and power consumption in a traffic-light control application, running on our *aaltOS* real-time kernel, in the 8-bit Atmel ATmega 2560 environment. Our study focuses on analyzing the relation between energy consumption and CPU usage of the application when operated at different clock frequencies and using different RTOS tick periods. Based on the results of the experiments, we discuss the usability of *aaltOS* for Energy Harvesting Applications.**

Keywords—RTOS, Power Consumption, Energy Consumption, Clock Frequency, System Tick, Energy Harvesting.

I. Introduction

Real-time operating systems act as a bridge between the physical and application layers in software hierarchy. The use of RTOS instead of a pseudokernel or traditional line-by-line programming improves the maintainability, usability, and portability of application code. As discussed in [1], an RTOS is usually installed on a device to provide an opportunity for multitasking and to ensure punctuality of deadlines. The popularity of using an RTOS in system development is growing, even in lightweight applications that run in 8-bit microcontrollers. This observation is supported, e.g., by the survey carried out in [2]; based on which it is not exaggerated to state that the use of RTOS in embedded systems provides efficiency in terms of power and energy consumption, system reliability, as well as response times.

Although RTOS development has been primarily focused on high-end processors (32-bit), there are several advantages of using RTOS for 8-bit systems, too. As discussed in [3], using an RTOS for smaller microcontrollers may strengthen the software development process, improve synchronization of various tasks, as well as intensify resource and timing management. The available research has clearly unmasked the use of RTOS in low-end computational devices, and reveals the motivation behind developing lightweight real-time kernels for 8-bit microcontrollers.

The concept of real-time systems demands the completion of tasks within specified periods, which is also a motivation behind the use of RTOS. Efficiency and usability of available RTOS can be analyzed by performance evaluation techniques as elaborated in [3]. The timing performance is certainly of no secondary value, but there is a growing need to consider power and energy consumption when selecting an RTOS, especially for remotely placed, battery-operated devices that do not provide easy access for frequent maintenance sessions. Timing performance and power/energy consumption vary obviously depending on the application and operating environment. The dependence of these performance measures on the clock frequency and system tick period may be of significant interest, too.

Fossil fuels have been the major source of energy for centuries. The conventional fossil fuel technique for extracting energy has its own pros and cons. The major advantage being large scale production of energy and the alarming con being rapidly decreasing amount of fossil fuel resources. With the advancement in sustainable and renewable energy fields, the concept of energy harvesting came into existence. The concept introduced recycling of various energy wastes like vibrational movements, thermal radiations and electromagnetic radiations to harvest energy. One out of many real-world examples is discussed in [4] where the authors have carried out a study to harvest energy from the electromagnetic waves transmitted between a multi-node network of transceivers. The significance of energy harvesting techniques can also be unmasked from its application in the internet of medical things field. The study carried out in [5] clearly depicts the concepts effectiveness in the field of wireless medical equipment and sensors.

In Section II, we introduce the *aaltOS* real-time kernel that forms the software platform for our demonstration application, which is described in Section III. Using a traffic-light control application for illustration, we reveal the effects of clock frequency and tick period on power and energy consumption of the system in Section IV. Our paper presents results from a case study carried out on the Atmel ATmega 2560 microcontroller. This study focuses on revealing the effects of changing clock frequency and system tick period on power and energy consumption using an Arduino platform. Finally, in Section V, based on the results of the previous analysis, we discuss the suitability of *aaltOS* for energy harvesting applications, and provide some concluding remarks

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

II. Real-Time Operating System's Features

A. *aaltOS RTOS*

aaltOS is a real-time operating system designed specifically for 8-bit microcontrollers and targeted for low-power energy harvesting systems. It is written in C language. *aaltOS* contains all the essential components that can be found in a typical real-time kernel. The main features along with the list of available API's falling in each category are listed below. For the sake of brevity of this paper, the argument list and return types associated with the API's are not listed.

- *Scheduler:* It is responsible for selecting and dispatching the task to be executed. *aaltOS* scheduler runs a preemptive-priority based scheduling algorithm. Tasks are assigned fixed priorities at their creation time.
- *Mutex:* The operating system provides an opportunity to individual tasks to lock shared resources. Mutex related API's supported by aaltOS are given below:
  - `aaltOS_mutexLock()`
  - `aaltOS_mutexRelease()`
- *Mailbox:* Used for inter-task communication and synchronization; tasks can send and receive messages using mailboxes. The mailbox related API options provided by *aaltOS* are listed below:
  - `aaltOS_mailboxPost()`
  - `aaltOS_mailboxPendMessage()`
  - `aaltOS_mailboxFull()`
  - `aaltOS_mailboxPopMessage()`
  - `aaltOS_mailboxGetNext()`
- *Event-based synchronization:* The tasks can also be synchronized using a lighter-weight (compared to mailboxes) method of event-based synchronization. A task can signal to other waiting tasks, which can then continue their processing. *aaltOS* provides the following API's in order to achieve event based synchronization between tasks:
  - `aaltOS_eventSignal()`
  - `aaltOS_eventReset()`
  - `aaltOS_eventBroadcast()`
  - `aaltOS_eventWait()`
- *Device Drivers:* Interface developed to operate and control external hardware devices attached to the controller. The list of API's provided by *aaltOS* in the category of device drivers is provided below:
  - `aaltOS_deviceOpen()`
  - `aaltOS_deviceClose()`
  - `aaltOS_deviceRead()`
  - `aaltOS_deviceWrite()`
  - `aaltOS_deviceIoctl()`
- *Interrupt Service Routine:* The operating system consists of an interrupt handler that executes an interrupt service routine when the controller receives a timer interrupt or a system tick.
- *Energy Saving Mode: aaltOS* is equipped with an energy saving mode that is capable of putting the CPU to sleep while idling—this is one of the core features needed in energy-harvesting applications.
- *Task Related API's:* In addition to the functionality related API's provided by *aaltOS,* the OS also offers the following API's for task related manipulations.
  - `aaltOS_createTask()`
  - `aaltOS_taskGetData()`
  - `aaltOS_taskSleepTicks()`
  - `aaltOS_taskSleepMilliseconds()`
  - `aaltOS_taskYield()`
  - `aaltOS_taskID()`
- *System Level:* System level API's available for *aaltOS* include.
  - `aaltOS_init()`
  - `aaltOS_start()`

B. *Application Programming Interface*

Following the illustrative API comparison shown in the study of [3], below (see Fig. 1) is a classification of available API's, or system services, provided by *aaltOS*.

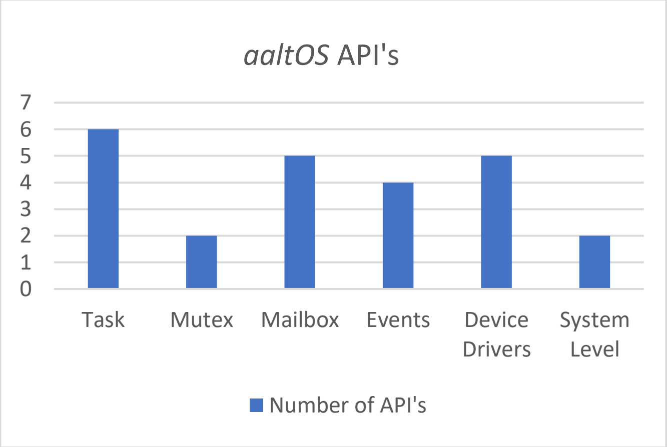


Fig 1. Number of *aaltOS* API's in different categories

C. *Memory Requirements*

For an application running on *aaltOS*, the total size of required data memory can be calculated using the following formula. Besides, the program memory required by the operating system is only 4554 bytes.

Data Memory (bytes) = Static data allocated by the user + Stack used by `main` before calling `aaltOS_start()` + 581 * N_t + 17 * N_d + 273

Where,

N_t = Number of tasks
N_d = Number of device drivers

III. Demonstration Application

A traffic-light control application was implemented using *aaltOS* for analysis purposes. The functionality was achieved using the Arduino Mega development board [6] coupled with a 7" TFT LCD Screen. The development board boasts an Atmel ATmega 2560 [7] microcontroller with a 16 MHz clock frequency and 16 ms tick period.

The complexity of this demonstration application resembles the one of a typical sensor network node that is powered by an energy harvester. And such an application is the target of *aaltOS*.

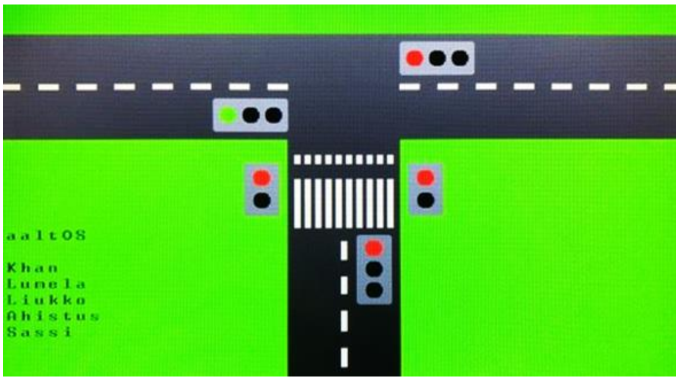


Fig 2. Application interface

This application models a T-intersection traffic-light system with additional pedestrian light crossing feature (see Fig. 2). The pedestrian request was generated using external push buttons. Under normal operation with no pedestrian crossing requests, three road lights switch their states periodically for the specified amount of time. In case a pedestrian crossing request is received, the normal operation is suspended and control is transferred to tasks controlling the states of pedestrian lights. After the designated amount of time is passed, the control again shifts to normal operation.

A. *Demonstration Application Tasks Structure (Table 1)*

1) Display Task: The functionality of this task is to properly initialize the LCD panel and draw the background. Another responsibility of this task is to update the display upon receiving update requests from the tasks controlling the states of the highway and pedestrian lights. The Display Task has its operations derived from the UTFT library for Arduino display control.

2) Lane Tasks: A total of three Lane Tasks exist for highway traffic lights. The responsibility of these tasks is to update the state of each traffic light depending on its previous state and the state of other contemporary highway traffic lights. After successful state updates the Lane Tasks are responsible for signaling to Display Task for corresponding display updates.

3) Pedestrian Task: This task is responsible for updating the state of the pedestrian lights depending on the input signal from the Polling Task. The state update is followed by a signal to the Display Task. The state of pedestrian light is updated based on the input from push button and previoous state.

4) Polling Task: The purpose of this task was to repeatedly check for the pedestrian request which mirrors to input signal from the push button. The Polling Task then signals the Pedestrian Task to update the states accordingly.

TABLE 1. APPLICATION SPECIFICATIONS

Number of Tasks	6
Program Memory Usage	7720 bytes
Data Memory Usage	4460 bytes
Required System Tick Period	Max. 16 ms

IV. Performance Analysis

In this section, we discuss the timing performance and power/energy consumption measures of the *aaltOS* real-time kernel. These timing performance measures are not application specific, but are rather linked directly to the operating system implementation, whereas the power and energy consumption values naturally vary from application to application.

A. *Timing Performance Measures*

1) Test Scenario: The testing was a two-step process. The hardware environment used for carrying out the tests consisted of the Arduino Mega development board running on 16 MHz clock frequency with *aaltOS* tick period of 16 ms.

a) Simulation: Using the on-chip simulator provided by Atmel Studio and setting up the operating conditions according to our application environment, we measure the actual number of clock cycles consumed by different operations.

b) Oscilloscope Measurements: Input/Output pins were manipulated to toggle a flag value and using an oscilloscope, the time difference was measured. The flag (GPIO pin) was set before calling the API under-observation and was reset after function execution. Repeated values were recorded and their mean was considered. The measured values were in accordance with the readings obtained from the on-chip simulator.

2) Test Results: The following timing results were obtained after carrying out the above-mentioned two-step process:

Task Creation Time:	100 µs
Task Switching Time:	50 µs
Event Signal Time:	100 µs
Event Wait Time:	200 µs
Mutex Lock Time:	9 µs
Mutex Release Time:	15 µs
Mailbox Post Time:	25 µs
Mailbox Pend Time:	50 µs

B. *Energy and Power Consumption Measures*

a) Test Environment: The test environment was created by triggering the CPU sleep/wake time during normal operation of the demonstration application. Employing similar flag set/reset mechanism as used for timing performance measurements, the total CPU usage was measured. Based on the Active and Sleep mode current consumed by the microcontroller, the power levels were calculated, and these were later used for energy consumption analysis. Energy and power consumptions were measured for different values of clock frequency and system tick.

The power levels on different values of clock frequency were calculated using the operating voltages and the amount of current drawn by the development board.

Active Power = Operating Voltage *Active Supply Current

Sleep Mode Power = Operating Voltage * Sleep Current

Due to the operation of the controller in two different modes, two different power levels were obtained depending on the current drawn by the microcontroller. To measure the energy consumption, operating modes of the microcontroller were recorded while running the traffic light control application.

The GPIO pin 13 of the Arduino Mega development board was set as soon as the controller attained Active mode and it was reset while the controller was operating in Sleep mode. The amount of time spent by the microcontroller in both of these operating modes was recorded, and using the specified power levels energy consumption was calculated.

Energy Consumed = Active Power * Active Time +
Sleep Mode Power * Sleep Time

b) Results: Fig. 3 shows the energy consumption (in mJ) of the microcontroller. It is clearly visible that decreasing the tick period increases the CPU usage, which in turn increases the amount of energy consumed due to the fact that the CPU spends less time in the Sleep mode (power consumption << 1 mW).

The default operating condition of the demonstration application demanded a tick period of 16 ms. Fig. 4 shows the effect of varying clock frequency of the microcontroller, while keeping the system tick constant.

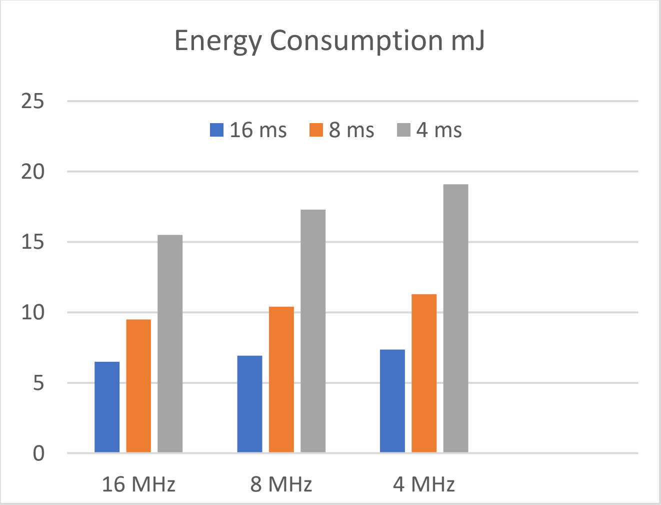


Fig 3. Energy consumption vs. clock frequency and tick period

- There was a predictable decrease in Active Power consumed, while moving from 16 MHz to 1 MHz, because of lower supply-current requirement of CMOS integrated circuits on lower clock frequencies.
- When the CPU usage was doubled, on each step moving from 16 MHz to 1 MHz, the behavior should be a constant level of energy consumption. Once we moved from 16 MHz to 8 MHz, the CPU usage doubled but the current stepping was not exactly halved (20 mA to 11.5 mA). This appeared to be the reason behind a slight increase in energy consumption on each step.
- One anomaly was observed from the Energy Consumption curve at the point of 2 MHz clock frequency. The energy consumption value at this point is 7.06 mJ, which is lower than the value at 4 MHz operating frequency. This behavior can be explained by looking at the Supply Current characteristics of the controller shown in Table 2 [7]. The Active Power and Energy Consumed recorded in Table 2 were measured following the approach discussed in Section IV B. The Supply Current was measured using the same method by noting the time spent by the microcontroller in Active mode. The ratio of current at 4 MHz to current at 2 MHz is 2.17, which is higher than any other successive current drop ratios, ranging from 1.71 to 1.77.
- It can be observed that the changes in energy consumption are negligible. From this, we can deduce the obvious fact that to do the same amount of work, we require the same amount of energy! The only thing that varies if we change the clock frequency is the rate at which the work is being done.

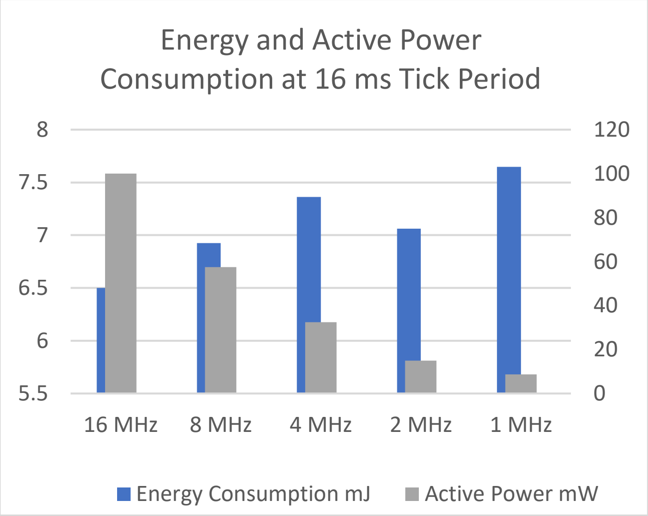


Fig 4. Energy and power consumption vs. clock speed

TABLE 2. PERFORMANCE MEASURES VS. CLOCK FREQUENCY

Clock Frequency MHz	CPU Usage %	Supply Current mA	Active Power mW	Energy Consumed mJ
16	0.03	20.0	100	6.50
8	0.06	11.5	57.5	6.92
4	0.12	6.5	32.5	7.36
2	0.24	3.0	15.0	7.06
1	0.48	1.8	8.8	7.65

V. Discussion

Energy-harvesting techniques are increasingly applied for running distributed sensor nodes. These systems can be installed in places that are not easy to access frequently, due to their geographical positioning. Such systems are usually made self-sufficient in terms of power generation, while the generated power can achieve maximum values of only a few milli-Watts [8]. Considering successful application examples from real-world scenarios, energy-harvesting techniques can be applied in wheel speed sensors of automobiles [10], wireless sensor networks [8], etc. Energy harvesters predominantly employ the use of kinetic, solar, and thermal sources to generate electrical energy locally. As stated earlier energy harvesters usually operate in low power level conditions. These power levels correspond to the low frequency operation modes used in experiments on traffic light control application. It is thus safe to say that such a low frequency operating mode when used in embedded devices which support energy harvesting can provide us with all the benefits of using operating systems as well as giving us the advantages of energy harvesting theory. As discussed in [9], the energy harvesting devices operate intermittent because of such low generated power. A solution proposed in the same study could be employed in lightweight operating systems such as *aaltOS* to get the best of both worlds. The solution suggests priority based scheduling of tasks. By modifying the operating system behavior by only a discrete amount such that only one task executes before the power runs out to minimize context switching overhead, we can get optimized energy consumption behavior from the embedded device under consideration.

The results discussed in the previous section reflect the microcontroller's energy and power consumption on different clock frequencies. Because of the intrinsic property of CMOS integrated circuits, a linearly varying trend in power consumption was observed. On the other hand, in order to achieve lower energy consumption values at a certain clock frequency, the tasks of the application should be optimized. Due to the fact that the amount of energy consumed even at low clock frequencies was almost equal to the energy consumed at higher frequencies, and taking into account the substantial decrease in Active Power consumption, we can state that the use of a small microcontroller running a lightweight RTOS, like our *aaltOS*, at low clock frequencies can provide us with all the required benefits of RTOS. Moreover, it can be used in energy harvesting systems to fulfill the power demands of lightweight embedded systems.

Acknowledgment

The authors are grateful to the anonymous reviewers for their constructive comments, which improved the manuscript significantly.

References

[1] M. T. Elsir, P. Sebastian, and V. V. Yap, “A RTOS for educational purposes,” in *Proceedings of the International Conference on Intelligent and Advanced Systems*, Kuala Lumpur, Malaysia, 2010, pp. 1–4.

[2] P. Hambarde, R. Varma, and S. Jha, “The survey of real time operating system: RTOS,” in *Proceedings of the International Conference on Electronic Systems, Signal Processing and Computing Technologies*, Nagpur, India, 2014, pp. 34–39.

[3] T. Nguyen, B. Anh, and S.-L. Tan, “Real-time operating systems for small microcontrollers,” *IEEE Micro*, vol. 29, no. 5, pp. 30–45, Sept.–Oct. 2009.

[4] Y. J. Kim, H. S. Bhamra, J. Joseph and P. P. Irazoqui, "An Ultra-Low-Power RF Energy-Harvesting Transceiver for Multiple-Node Sensor Application," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 11, pp. 1028-1032, Nov. 2015.

[5] Y. Rajavi, M. Taghivand, K. Aggarwal, A. Ma and A. S. Y. Poon, "An energy harvested ultra-low power transceiver for Internet of Medical Things," ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference, Lausanne, 2016, pp. 133-136.

[6] Arduino, *Arduino Mega*. [Online]
Available: http://www.arduino.cc/en/Main/arduinoBoardMega

[7] Atmel Corporation, *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V Data Sheet*. [Online]
Available: http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[8] T. J. Kaźmierski and S. Beeby (Eds.), *Energy Harvesting Systems: Principles, Modeling and Applications*. New York: Springer, 2011.

[9] C. Pan, M. Xie and J. Hu, "Maximize energy utilization for ultra-low energy harvesting powered embedded systems," 2017 IEEE 23rd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hsinchu, 2017, pp. 1-6.

[10] D. Parthasarathy, “Energy harvesting wheel speed sensor,” Master Thesis, Chalmers University of Technology, Göteborg, Sweden, 2012. [Online]
Available: http://publications.lib.chalmers.se/records/fulltext/159168.pdf

Valuation of Intellectual Property – Income Approach and Scenario Analysis

Software on Early Stage of Implementation

Paweł Kossecki
The Polish National Film, Television and Theater School
ul. Targowa 61/63
90-323 Łódź, Poland
kossecki@poczta.onet.pl

Łukasz Korc
Management Faculty of Warsaw University
ul. Szturmowa 1/3
02-678 Warszawa, Poland

Stefan Kossecki
Kossecki Tax Planning
ul. Królewicza Jakuba 1A
02-956 Warszawa, Poland

The development of sophisticated software products is often related to high uncertainty of the effect. One of the important tasks of decision makers is proper assessment of the future financial value of the effect. A proper method for calculating the value of intellectual property is a crucial topic to make the proper decision.

Selected problems related to valuation and defining of intellectual property were described. Authors have developed a model of business valuation software on early stage of development.

Article describes advantages and disadvantages of common method and shows a case study of the valuation of intellectual property. Proposed model is based on scenario analysis.

Paper presents how to adjust common method of valuation for valuation of intellectual property related to software in case of lack of market comparisons.

Keywords — Software, valuation, technology valuation, intellectual property, copyrights, drones

I. Introduction

The growth of the American economy till the so-called Internet bubble has exceeded the wildest expectations of economists. It was largely associated with the development of information technology (IT) sector and fall in prices within IT equipment and telecommunication services in the last decade of the 20th century. The primary source of this economic growth was a decline in prices of semiconductors, which translated directly into product prices based on semiconductor technology, such as computers and telecommunication equipment, and indirectly also impacted on other sectors of the economy [5], [6], [10]. At the same time, new electronic products appeared on a massive scale, like laptops, mobile phones, PDAs, MP3 players, and many other that changed everyday life [10].

In the new economy, the professionalism and effectiveness of project implementation determines the competitive position of the company, its innovation, efficiency, effectiveness in the implementation of cutting-edge solutions, and customer perception of the company as an attractive service provider.

Together with the changes introduced by IT technologies to the traditional economy, a new economy appeared, referred to as the e-economy, e-marketplace and e-business. The term "new economy" emerged in the 1990s. It was associated with the changes IT technologies introduced to the traditional economy. The speed of the development of the sector of the new economy on one hand, and uncertainty on the other hand, were associated with the highly speculative attitude of stock-market investors and the fast development of Venture Capital and Private Equity funds [10].

Development of computer technology was largely based on introduction of new types of computer software, which can be considered as one of the types of intellectual property (IP). The development of a proper method for calculating the value of different types of intellectual property is an important task in the current climate of the new economy. It's important for the investment officers, as well as other decision makers to help in assessment financial value, as well as way of development of the new technologies. Current paper shows case study analysis of valuation software for drone control. Method is based on scenario analysis and DCF valuation.

II. Case Study Overview

Authors prepared valuation of software for drone control. Software was prepared by the scientific institute (SI). Drones will be in the future produced by the independent production company. Drone can be used for military purposes. Civil adoption is not possible. Software is ready to use solution and meets criteria of the Ministry of Defense (MD).

This case was described to show ability of valuation in case of lack of market and possibility of using market comparisons. There is no market for this type of software and producer of

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

drones must acquire copyrights to the software and pay for it fair price. Usually there is limited market and kind of monopoly for the owners of IP, in this case it's very evident.

Software can't be used for the civil purposes and sold to the third parties.

Ministry of Defense has opened a tender. Army wants to buy 500 drones over next 3 years. SI has developed basic software for controlling military drones. Together with company Drone Control ltd. (DC) institute is going to take part in the tender. However further research needs to be done in order to meet all criteria's described in tender's documentation. There are following scenario considered:

A. *Scenario 1* – MD chooses different suppliers of drones;

B. *Scenario 2* SI and MD will be one of the suppliers chosen by MD, but there is no further development of drone software needed;

C. *Scenario 3* SI and MD will be main supplier of the drones but minor changes to the software are needed;

D. *Scenario 4* SI and MD will be sole supplier of drones over next 3 years. But relevant development is needed.

III. Definition and Methods of Intellectual Property Valuation

In modern economy IP is one of the most important factors correlated with development of the country. F. ex. the higher number of patents economy has, the higher development of the economy. [16]

Intellectual property is either product, work or process invented by the company that can give it competitive advantage. There are three types of IP (T. Copeland) [1] [2] [15], [16]:

- industrial property (e.g. inventions, patents, trademarks, industrial design, geographical indications of origin);
- artistic work protected by copyright (e.g. artistic works, music, television broadcasts, software, databases, architectural design etc.);

commercial strategies (trade secrets, know-how etc.).

Computer software can be considered as one of the types of intellectual property. To consider it as a value for the owner, which can be valued and sold, there should be met the following conditions [11]:

- it must be protected,
- commercial applications must be identified,
- it must be useful for generating future incomes or cost cutting.

It should be possible disconnecting IP from the company's assets and carrying out the purchase / sale or other change control of future economic benefits generated by the possession of IP.

There are following methods of protecting intellectual property:

- trade secret,
- patent protection,
- copyrights.

Computer software is protected by the copyrights. There are the following common methods of the valuation of copyrights [3], [9], [14], [17]:

- market comparison,
- income approach,
- cost approach.

A. *The cost approach method*

The cost approach is based on the valuation of assets, based on the cost of their creation. This approach is usually not useful to analyze the problem, because in the case of copyrights there is a limited correlation between the cost of asset creation and the level of income generated by the assets [10].

B. *The market comparison method*

The market comparison assumes that free market supply and demand forces will create the so-called equilibrium price. In cases of valuing copyrights, methods based on income and market comparison are preferred, however, it should be taken into consideration that there exists only a limited market in some cases. Owners of IP often have a natural monopoly.

C. *The income approach*

The income approach is based on the valuation of assets for their ability to generate future income. The most popular is the Discounted Cash Flow method (DCF), based on the forecasting of free cash flow generated by the assets. Related methodology is based on the 25 percent *Rule of Thumb*.

Common equation for calculating value of the enterprise according to the income approach (DCF method) can be described as follows [1], [3], [7], [10]:

$$V_{DCF} = \frac{FCF_1}{1+r} + \frac{FCF_2}{(1+r)^2} + ... + \frac{FCF_t}{(1+r)^t} + \frac{TV}{(1+r)^t} \tag{1}$$

where:

V_{DCF} — enterprise value calculated according to the DCF method,

FCF — free cash flow,

TV —residual value,

r – discount rate.

Free cash flows can be calculated according to following equation [1], [11]:

$$FCF_t = NOPLAT_t + DEPR_t - CAPEX_t - \Delta WC_t \qquad (2)$$

where:

$NOPLAT$ – net operating profit less adjusted tax,

$DEPR$ — depreciation,

$CAPEX$ — capital expenditures,

ΔWC — changes in working capital.

For calculating value of the intellectual property based on income approach following equation must be modified to the following formula:

Cumulated cash flow of the owner of the right of intellectual property can be described with the following equation:

$$V_{IP} = \sum_{t=o}^{n} m \frac{NOPLAT(IP)_t + DEPR_t - CAPEX_t - \Delta WC_t}{(1+r)^t} + \frac{TV}{(1+r)^t}$$

(3)

where:

V_{IP} – value of intellectual property,

NOPLAT(IP) – expected net earnings acquired during commercial usage of evaluated technology rights (e.g. production of drones);

m –coefficient of division of profit between licensor and licensee,

n – expected lifetime of the intellectual property,

TV –residual value of the project.

D. Intangible assets valuation

When it comes to valuation of intangible assets, income approach is preferred with market approach following, especially by the organizations like OECD and tax authorities [10], however in case of military technologies we can hardly speak of free/not controlled market, as most of both producers and receivers of the technology are state controlled parties.

In case of market approach, market's practice has to be verified with comparative analysis method including such factors as: profitability, characteristic of the markets, purchasing power of final clients.

However, in analyzed case, due to lack of proper market data, and to small number of transactions that can be compared, market approach can't be used.

At the same time, we can assume that benefits of the buyer will be connected with future cash flow. Cash flow would be a result of both future license and expected income of implementing the product, sale income or sale of the license to the third party.

In analyzed case it's possible to state partition of financial benefits between parties – owner and user of the intellectual property. Usage of the rule *Arm's length* is accepted. It means that transaction is based on transaction conditions which would have been used by parties which are not related. Dividing benefits of intellectual property would involve cause and effect method. Both cost, transaction risk and function of the parties would have to be taken in consideration.

In case of calculating intellectual property value after the prediction timeline, residual value of the project has to be shown.

Described income approach method is consistent with the method of transaction profitability used to determine transfer pricing between related parties.

E. Discount rate

Discount rate used in equation above can be translated as a cost of own capital required by the investor, which he would have invested in a project with similar risk level. There is no single universal method of calculation cost of such capital [6], [8], [9]. It can be identified as highest possible return rate, which could be expected from investment in securities with same risk level. Several factors can affect cost of the capital [7]:

- How the funds are used?
- Risk of loss of liquidity of the company implementing the project;
- Possible negative change of market conditions;
- Change of market interest rate;
- Problems with acquiring funds for further development;
- Underestimating operating and investment costs;
- Lack of experience in similar projects;
- Other operation and investment risks.

In case of projects with low risk of abandoning the project, connected with core company activity, and for companies with stable market position, and for products with secured market sale, it's possible to estimate required return rate with weighted average cost of capital (WACC) for the company.

In many cases however, investors accept higher risk of investment in intellectual property. In exchange they are expecting higher rate of return. In this case we can use rate of return for Venture Capital funds as a benchmark.

Below are samples of return rates levels required by VC while investing in intangible assets [8], [9], [13]:

- start-up: 50%;
- first stage of development : 40%;
- second stage of development: 30%;
- Third stage of development: 25%.

Startup - company with idea and nothing more. Highest risk rate. Funds are used to perform basic research and building prototype of the product. Income is not crucial at this stage.

First stage of development - company has already prototype of the product, but further development needs to be done before mass production. Positive cash flow is expected inside a few years.

Second stage of development: company has successfully introduced product to the market, but further market expansion requires substantial funds, which might not be financed by the bank. Ability to create profits is proven, but further expansion needs higher funds than those provided by current activity.

Third stage of development - financing becomes unstable due to fast expansion of the company. Access to bank credits or stock exchange is limited. High and stable profits are possible, but additional funds are needed in order to expand worldwide.

Based on above information, in opinion of the person responsible for evaluating the company, in analyzed case study discount rate of the first stage of development should be used. - 40%

This assumption is based first of all on the early stage of development of the product and high risk connected with not meeting goals of sales level.

In order to evaluate value of the research and development, income method has to be used. It's result of limited access to official data of the transaction involving sale of license in drone technology.

Applied method (allocation of profit) is based on valuation of assets based on their future ability to create income. In order to use income method together with discounted cash flow (DCF), model of cash flow needs to be created. This model should allow to evaluate:

- Total income which is can be achieved by selling the product, where results of the research are used;
- Income distribution throughout the years;
- Operating margin obtained on the production of final product;
- Division of the margin between licensor and licensee;
- Residual value of valuated intellectual property;
- Cost of capital in similar operations.

Period of technological usefulness it's time from beginning of mass production until it's end. It's result of several factors f. ex. producer strategy, material availability, market changes or technological progress which demands replacing older technologies with new.

Based on model shown above it's possible to estimate added value for investors/shareholders which can be obtained with knowledge and materials created during research and development project.

By applying standard profit sharing ration between the licensor and licensee, you can estimate what part of profit should go to each of the parties.

Future incomes of the R&D results are estimated in advance, at the time of creating the evaluation. There is high risk that business assumptions will not meet expectations. Consequently, it's necessary to discount future profits (cash flows) of the project. Discount rate has to take in account risks of failure of the project, advance level of the project and all other factors, that may affect future success of the project. The higher risk of failure, the higher discount rate should be used.

At this stage, ownership structure and completeness of completed R&D work and not take into consideration, it means that we should make all calculation for whole documentation, which is needed to start production of final product.

Value of the license fee, which would be obtained by the owner of the R&D results, is equal to cash flow, calculated on the basis of applied model, after using proper discount rate and proper division ratio between licensee and licensor.

Next, completeness of the work and ownership structure must be taken in account.

Proportion of profit sharing between licensee and licensor

Proportion of license fee and margin can be assessed based on [14]: Royalty Rates for Licensing Intellectual Property, John Wiley & Sons, pp. 50. For the software products level is 21,4% of operating margin generated on the product.

If the licensor in order to achieve planned margins, uses several licenses. Then the license fee should be divided in proportion to value generated by technology that license provides.

V VALUATION – CASE STUDY ANALYSIS

Valuation below is based on described above methodology and case study of drone software presented in chapter II of the paper. There are assumed R&D costs of software project are following:

- Software development costs: 1.000.000 EUR (25% of total R&D costs);
- Expected continuation costs: 3.000.000 EUR (75% of total R&D costs).

Scenario analysis is presented in tab. I. Analysis is based on scenarios described in chapter II.

TABLE I. SCENARIO ANALYSIS

	Probability of Sacenario	**Number of Sold Drones**	**Unit price of Drone (EUR)**	**Sales (EUR)**	**Margin (EUR)**	**Cost of Project Continuation**
Scenario A	70,00%	0	0	0	0	0
	Probability of Sacenario	**Number of Sold Drones**	**Unit price of Drone (EUR)**	**Sales (EUR)**	**Margin (EUR)**	**Cost of Project Continuation**
Scenario B	10,00%	10	30 000	300 000	15 000	0
Scenario C	10,00%	90	30 000	2 700 000	15 000	100 000
Scenario D	10,00%	500	30 000	15 000 000	15 000	500 000

	Expected Level of Sales	**Expected Value of Margin**	**Expected Number of Drones Sold**	**Expected Manufacturing Cost**	**Expected Value of Continuation Cost**
Scenario A	0	0	0	0	0
Scenario B	30 000	1 500	1	28 500	0
Scenario C	270 000	1 500	9	268 500	10 000
Scenario D	1 500 000	1 500	50	1 498 500	50 000

Expected Value	**1 800 000,00**	**4 500,00**	**60,00**	**1 795 500,00**	**60 000,00**

Sales Forecasts based on scenario analysis are presented in tab. II. First delivery is planned in 2020. Forecasts are based on data presented in tab. I. All delivery should be completed before end of 2023.

TABLE II. SALES FORECASTS

	Total	**2018-2023**	**2018**	**2019**	**2020**	**2021**	**2022**	**2023**
Planed # of Drones Sold		240			60	60	60	60
Revenues from Sales of Drones	EUR	1 800 000			450 000	450 000	450 000	450 000

Based on presented scenario analysis and assumptions valuation of IP related to R&D software was prepared and data are presented in tab. III. Valuation date is 01/01/2018. Effective corporate income tax on the level 17,04% was assumed. Proportion of license fee and margin was established on the level of 21,4% of operating margin generated on the product (software) [14]. Discount rate is 40% (first stage of development.

TABLE III. VALUATION OF INTELLECTUAL PROPERTY

		2018	**2019**	**2020**	**2021**	**2022**	**2023**
Sales Forecast	000 EUR			450 000	450 000	450 000	450 000
Manufacturing Costs of Drones	000 EUR			448 875	448 875	448 875	448 875
Operating Margin of Manufacturer	000 EUR			1 125	1 125	1 125	1 125
% margin on production	%			0,25%	0,25%	0,25%	0,25%
Division of Profit between Licensor Licensee	%			21,40%	21,40%	21,40%	21,40%
Total Value License Fee R&D Work	000 EUR			241	241	241	241
		2018	**2019**	**2020**	**2021**	**2022**	**2023**
% of Valuated R&D Work in Total R&D Costs	%			25,00%	25,00%	25,00%	25,00%
License Fee Related to Valuated IP	000 EUR			60,19	60,19	60,19	60,19
Effective Corporate Income Tax Rate	%	17,04%	17,04%	17,04%	17,04%	17,04%	17,04%
Cost of Capital for Companies on 1st stage of Development	%	40,00%	40,00%	40,00%	40,00%	40,00%	40,00%
Discount Rate		1,96	1,40	1,00	0,72	0,51	0,37
Tax Shield	000 EUR			10,26	10,26	10,26	10,26
Discounted License Fee (FCF)	000 EUR			70,57	50,41	36,01	25,72
Total Value of R&D Work Related to Drone Software	**000 EUR**	**182,71**					

Calculated value of intellectual property is 182.710 EUR.

Discussed case shows that at this point, development of drone software is connected with high risk of failure and at the same time return is at the average level. This may indicate need of public investments in this type of high risks projects or cooperation of public institutions and private entrepreneurs.

VI CONCLUSIONS

Described method was used by one of author to assessment value of IP related to software project on early stage of implementation.

Method is useful mainly, because valuation is prepared on public data and outcome is related to economic profits of licensee using the IP.

Calculated value show high risk and uncertainty of the project, as well as low value of IP in comparison to the amount of money spent to the project.

This type of projects usually is realized not only for commercial purposes, but mainly for realization of public strategies. As well as building structures and know-how of development of country high tech industry. Which might become sources of further development of the industry.

REFERENCES

[1] T. Copeland, T. Koller, J. Murrin Valuation: Measuring and Managing the Value of Companies, John Wiley & Sons, New York, 1990

[2] A. M. Dereń „Zarządzanie własnością intelektualną w transferze technologii" Diffin, Warszawa 2014

[3] European Commission, "Growth, Internal Market, Industry, Enterpreneurship and SMEs, https://ec.europa.eu/growth/industry/intellectual-property_en, 2017

[4] D. Frykman, J. Tolleryd, Corporate Valuation: an easy guide to measuring value, Prentice Hall, New York, 2003

[5] K. Gilbert, “The Valuation of Copyright-Related Intangible Assets, Insights”, Willamette Management Association, Autumn 2009, pp. 35-46

[6] D. W. Jorgenson, Information Technology and U.S. Economy, Harvard Institute of Economic, Research Discussion Paper, 2001

[7] D. W. Jorgenson, K. J. Stiroh, “Information Technology and Growth”, American Economic Review, vol. 2, pp. 109–115, 1999

[8] P. Kossecki, Wycena i budowanie wartości przedsiębiorstw internetowych, Wydawnictwa Akademickie i Profesjonalne, Wyższa Szkoła Przedsiębiorczości i Zarządzania im. L. Koźmińskiego, Warszawa, 2008

[9] P. Kossecki, „Kreowanie i pomiar wartości przedsiębiorstwa w świecie Internetu”, PWSFTViT Publishing House, Łódź, 2011

[10] P. Kossecki, „Metodologia określania wartości praw autorskich - kalkulacja odszkodowań do celów prawnych”, E-Mentor, nr 3(45), pp. 46-50, 2012

[11] P. Kossecki, “Valuation and Value Creation of Cable TV Operators. Cable Operators and Copyright Fees”, PWSFTViT Publishing House, Łódź, 2015

[12] OECD, OECD Transfer Pricing Guidelines for Multinational Enterprises and Tax Administrations 2017, http://www.oecd.org/tax/transfer-pricing/oecd-transfer-pricing-guidelines-for-multinational-enterprises-and-tax-administrations-20769717.htm

[13] R. Razgaitis, “Valuation and Dealmaking of Technology-Based Intellectual Property”, dr Richard Razgaitis, 2009

[14] G. V. Smith, R. L. Parr, “Intellectual Property, Valuation, Exploitation, and Infringement Damages”, John Willey & Sons, Hoboken, 2005

[15] K. Szczapanowska-Kozłowska „Własność intelektualna : wybrane zagadnienia praktyczne” LexisNexis, Warszawa 2013

[16] R. Taplin and A. Z. Nowak “Intellectual property, innovation and management in emerging economies” Abingdon, Routledge 2010

[17] D. Zarzecki, „Metody wyceny wartości niematerialnych i prawnych”, e-rachunkowość, http://e-rachunkowosc.pl/artykul.php?view=404&part=2, 23.11.2011

Data Analysis Tool Supporting Software Development Process

Rafał Kozik
Institute of Telecommunications and Computer Science
UTP University of Science and Technology,
Bydgoszcz, Poland
rkozik@utp.edu.pl

Michał Choraś
Institute of Telecommunications and Computer Science
UTP University of Science and Technology,
Bydgoszcz, Poland
ITTI Sp. z o.o, Poznań, Poland
mchoras@ittic.com.pl

Damian Puchalski
ITTI Sp. z o.o, Poznań, Poland
dpuchalski@itti.com.pl

Rafał Renk
ITTI Sp. z o.o, Poznań, Poland,
Adam Mickiewicz University, Poznań, Poland

***Abstract*— As rapid software development gain in popularity, the complexity of project management and quality assurance has dramatically increased. This has led to traditional approaches to quality management becoming ineffective. On the other hand, technologies supporting development process have also improved. Recently, automation tools for software quality checking have given project managers a wide spectrum of additional solutions for monitoring the developed product. In this paper, we present methodology and tools for gathering and analyzing data related to software development. Our goal is to improve rapid software development process with machine-learning, data analysis and visualization techniques. In this research, we gathered data from various sources. In particular, we hypothesize that process related aspects such as planning, testing, and bug tracking can be correlate with metrics related to software quality. Results are promising and show that the proposed solution can be a useful tool to improve quality of developed software.**

Keywords—software quality, development process, project amangement tool, data mining, and pattern analysis

I. Introduction

With the recent growth of data volume observed in a number of areas and still evolving phenomenon of big data, terms such as data mining, data analytics and data science are become more and more popular and get increasing attention of scientists and business. Data science is referred to the multi-disciplinary field including scientific methods, processes, and tools applied to extract knowledge from (often big) data related to given context. Data science combines statistics and data analysis methods to understand actual or past phenomena based on the analyzed data. One of the data analysis methods widely employed to analyze big data is e.g. machine learning, applied for example in cyber security to build models of network, to detect anomalies/intrusions, or for risk management and forensics purposes. However, in this paper, we focus on relatively new paradigm, namely employing data science and machine learning for estimation and optimization of software development quality. From the one hand, cost of quality assurance and testing in IT are growing from year to year. Currently, IT organization spent approximately 1/3 of their budgets on quality assurance with the trend of rising this value to about 40% in next three years [1]. From the other hand, low quality of developed solutions, including low quality of code, significantly impacts the overall cost of the software development, deployment and further maintenance [2]. According to [3], the process of debugging software during its design phase costs 4 to 5 times less than fixing bugs after its release. Also, there is an estimation that newsworthy software failure declines its stock price with an average of 4% to 6% (for companies experiencing multiple software failures), what generates almost 3 Billion Dollars of market losses [3]. It is a quantitative rationale for the development of innovative, data science-driven means to ensure preventive and proactive quality assurance in the context of software code, rather than focusing on technologies and processes for reactive quality assurance.

The ideas presented in this work are the results of our cooperation with H2020 QRapids project [4]. The main pillars of our research are built on top of the experiences in data mining, cyber security [5] and critical infrastructures protection areas [6].

This paper is structured as follows: system use cases, actors, information flow and proposed architecture are presented in section 2, our experiments goals are shown in section 3, results of experiments related to correlation of software development quality metrics and their prediction are presented in section 4, findings and observations drawn from the experiments are presented in section 5. The paper is concluded with final remarks and plans for future work.

II. Proposed System Architecture

In this section, the architecture of the proposed system is presented. First, we demonstrate the conception for the information flow used for the data analysis. Afterwards, the

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

TABLE I. CATEGORIZATION OF METRICS USED FOR DATA ANALYSIS

Category	*Type*	*Metric or information*
Source code	Code history	Commits – number, date, detailed description
		Developers – number, names, and involvement
		Branches – names, commits, changes history
	Code Quality	Complexity of classes, functions, and files
		Number of duplicated lines and its density
		Comments Density and its number
Development process	Testing	Testing time – maximum, minimum, and average
		Passed failed tests indicated for specific functionality, feature or improvement
	Backlog	Features and task planned for sprint backlog
		Time spent to implement specific feature or functionality
	Issues	Number of issues currently opened and recently closed
		Open issues – these bugs, tasks and features that remain unsolved
		Re-open issues – those which has be opened back due to some improvements of modifications
	Bugs	Number of bugs reported to specific functionality or feature
		Bugs criticality
		Time to fix a specific bug
Running Application	Usage	Time spent on using particular features (average, max, and min)
		Used features – list of most frequently used functionalities
	Security	Vulnerabilities indicated in the code
		Exploits that have been reported
		Criticality of security flaws
	User Feedback	Rates given by users to evaluate usefulness of the application
		Comments posted by the users

information data sources and key system elements are briefly presented.

A. The system use cases and actors

We believe that the presented in this paper research can evolve toward the advanced system that will help in meeting quality requirements, better plan tasks and issues, better assign tasks to particular developers, and finally, meet strategic indicators of the product and organization (such as time to market, etc.). We also believe that this approach is in align with QRapids project activities, which ultimate goal is to support the process of rapid software development with an increased awareness of software quality matters [7, 8].

The system proposed in this paper system could be devoted to middle-level roles in the organization, such as product owners, SW team leaders, scrum masters etc. To keep this paper self-containing we introduce following terms that are further used in the paper, namely:

- Issue – as an unit of programming work needed to accomplish some defined progress of software development (e.g. performing test, implementing given feature, fixing bug, etc.),
- Task – is the small (undividable) portion of programmer's work that leads to solve certain development problem,
- Sprint – is the period of time in which programmer's team has to complete specific tasks, after the sprint completion the results have to be ready to review. Each sprint starts with the sprint planning,
- Code quality metrics – are measurable values referred to the certain aspect of developed code (e.g. code complexity, code repetition),
- Strategic indicators – are aggregated information for the decision makers estimated based on the quality metrics and related to the quality requirements.

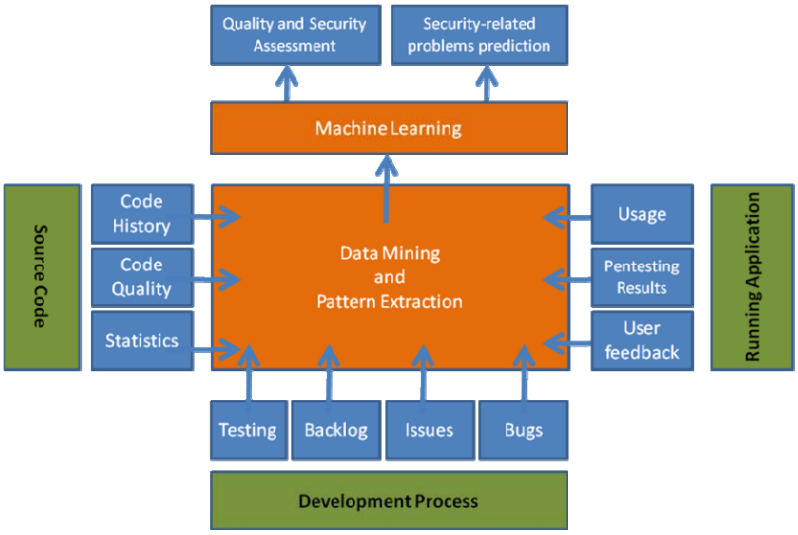


Fig. 1. The conceptual architecture of the proposed solution. Currently, we use three key data sources for our analysis: source code, development process data, and data collected from running application.

B. The information flow

The conceptual architecture of the system is presented in Fig. 1. We use several data sources to measure the statistics (we call these metrics) related to the project. These metrics are retrieved from GitLab and SonarQube project management tools. In the future, we plan to extend this list, since in different organizations and software houses, different tools are used (e.g. some teams/organizations may use Jenkins and some GitLab CI instead [13]). However, in many cases, it is just a matter of having the right connector between a project management tool and our prototype. Some examples of these metrics are included in Table 1. It must be mentioned that the first and the second category of metrics can be obtained from GitLab and SonarQube tools. The third category is a plan for future..

The collected data is stored in our system for further processing. For example, the project manager has the ability to access a variety of data in one place and use additional functionalities such as visualization, correlation analysis (both presented in the experiments section), and prediction of quality metrics. However, in this paper, we particularly focused on a correlation analysis to find relevant relations between the metrics and proof some of our research hypothesis.

C. The architecture of the system

The current architecture and deployment model is presented in Fig. 2. There are three key elements namely: our proxy, Apache Kafka [9], and Apache Spark [10]. The proxy implements the interface to the external project management tools (GitLab [11] and SonarQube [12]). It is responsible for connecting (e.g. via VPN [14]), downloading the data, preliminary processing, and removing sensitive data. The proxy also provides Graphical User Interface (GUI) for data visualization and system configuration.

The Apache Kafka publish-subscribe system is used to communicate with Apache Spark. In particular, the preprocessed data is published at a specific Kafka topic and further consumed by the Apache Spark framework, where complex and more sophisticated data processing patterns can be used.

A proxy design pattern for data gathering has been used in the proposed architecture due to the privacy reason. When we consulted this architecture with our possible end-user, we learned that customers usually want to have control over data

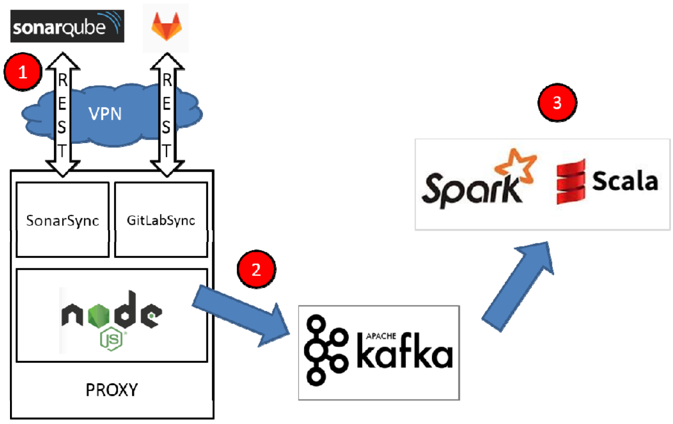


Fig. 2. Information flow used by the proposed solution. The project-related data is collected via VPN (1) at the proxy instance (2) where basic data preprocessing, analysis and anonymisation take place. Computationally expensive operations, data mining and machine learning take place in the apache cluster (3).

pushed to a cluster or a cloud.

The described architecture was deployed and used in the experiments (based on real life data from software development companies) described in Section 3.

III. Experiments

For the experiments, we have used real life data collected while developing real commercial products for customers at software development companies. The companies used GitLab tool to manage the project-related data, namely issues (backlog, user stories, features, tasks, and bugs), source code repository, and continuous integration (CI). To control the quality of the produced code SonarQube tool has been used.

The goal of our experiments was to validate the correctness of the architectural assumptions, to verify the research hypotheses related to correlation of software quality metrics with the process management characteristics, and to assess the usefulness of provided functionalities.

The data from GitLab and SonarQube tools were collected incrementally as received. The details on the collected data and tentative results have been presented in the next section.

IV. Tentative results

In this section, we have presented some tentative results obtained for the proposed system. First, we demonstrate the graphical user interface that gathers all project-related data in one place. Afterwards, we present tentative experiments related to the effectiveness of metrics prediction.

A. Graphical User Interface

The Graphical User Interface allows the project manager to have all the data in one place. Moreover, additional visualization features allow for presenting characteristics that are not available in the GitLab. Some of these additional features include a number of opened features changing over time (example shown in Fig.3), sprint backlog size, number of bugs, and number of tasks being in progress.

Also in case of SonarQube, we have provided additional functionalities that are not available by default in the original tool. For example, we use a variety of figures to present code related characteristic for changing evaluation periods. By default, SonarQube presents only the most recent ones.

B. Metrics cross-correlation and prediction

In our experiments we used gathered data from the ongoing software development process to verify following hypothesize:
H1: Increasing size of sprint backlog correlates with the increasing cognitive complexity and amount of duplicated lines, as the result of work under the time pressure

H2: Increasing complexity of code correlates with increasing number of defects (bugs)

H3: Increasing amount of duplicated code correlates with increasing number of defects (bugs)

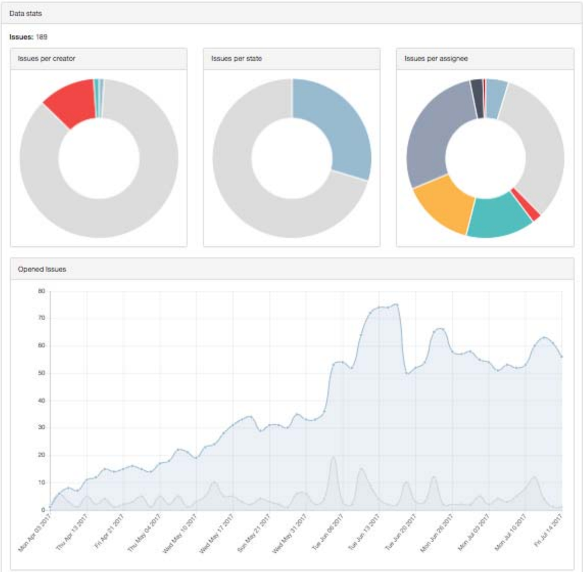


Fig. 3. The example of figures generated for data obtained from GitLab. Upper row: number of issues per creator (left), per type (middle), per assignee (right). Bottom row: Number of open issues vs time (upper line) and a number of new issues vs time (bottom line).

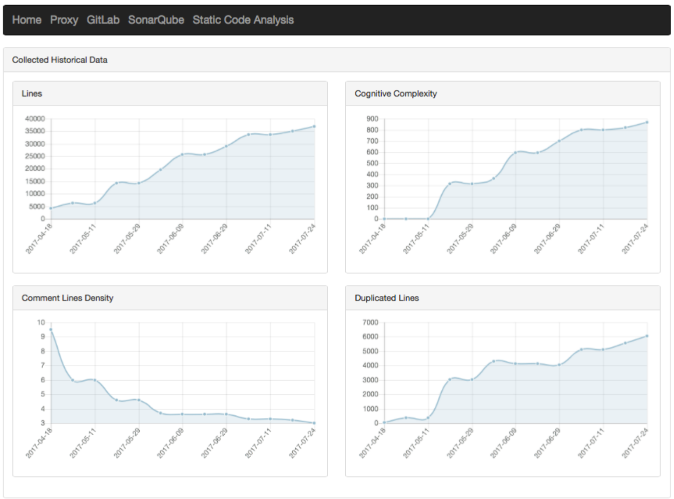


Fig. 4. Selected metrics obtained from SonarQube, measured over the project lifetime. Upper row: number of lines, cognitive complexity. Bottom row: comments density, number of duplicated lines.

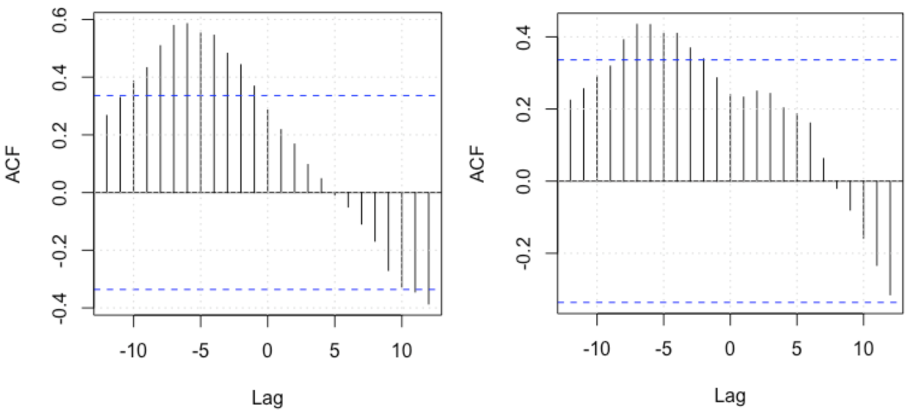


Fig. 5. The Correlation (ACF [15]) between the backlog size and the cognitive complexity for varying lag (left) and the backlog size and the number of duplicated lines (right). The dashed lines represent an approximate confidence interval (95%).

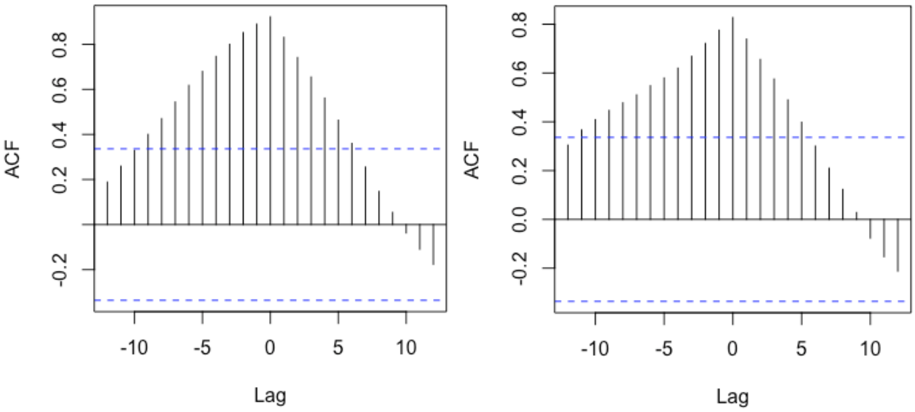


Fig. 6. The Correlation (ACF) between the cognitive complexity and number of bugs (left) and the number of duplicated lines with number of bugs (right). The dashed lines represent an approximate confidence interval (95%).

To verify these hypotheses, we have used estimates obtained from cross-correlation function of two time series. The results verifying the two first hypotheses are shown in Fig. 5. It can be noted that there exist statistically significant correlation between the size of the sprint backlog and the cognitive complexity of code as well as the number of the duplicated lines. For both of these metrics we have observed higher correlations for negative lags. It means that the decreased code quality in the future will be result of the currently oversized backlog. This could be a valuable information for the product manager, which can help plan the future work.

The estimates of cross-correlation function for the third hypothesis are included in Fig.6. It can be noted that these values are high and statistically significant. However, this relation is not surprising, since usually complex code is prone to errors, in particular when time pressure is presented in the development process.

V. Conclusions

In this paper, we presented the tentative results obtained for the proposed approach to software quality evaluation. In the proposed research we have combined information from two independent source. The development and process related data have been collected from GitLab tools. The software quality metrics have been calculated (and monitored) with SonarQube system. We have validated our research hypotheses using the realistic use-case, which concerned a development process of a web-application system. The obtained results are promising and in our opinion can serve as a starting point to further research. Our plans for future work include more sophisticated cross-correlation and prediction scenarios. In particular, we plan to use the proposed framework to detect possible cyber security flaws.

Acknowledgment

This work is funded under BSM 80/2017 and Q-Rapids project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732253.

References

[1] Capgemini: World Quality Report 2016-17, Eighth Edition. Available: https://www.capgemini.com/world-quality-report-2016-17/

[2] Jones, Capers, and Olivier Bonsignour. The economics of software quality. Addison-Wesley Professional, 2011

[3] QASymphony, The Cost of Poor Software Quality, Available: https://www.qasymphony.com/blog/cost-poor-software-quality/

[4] H2020 project Q-Rapids, http://www.q-rapids.eu/

[5] Choras M., Kozik R., Puchalski D., Holubowicz W., Correlation approach for SQL injection attacks detection, 7th Int Conf on Soft Comp Models in Industrial and Environm Applications/5th Computational Intelligence in Security for Information Syst/3rd Int Conf on EUropean Transnational Education, 177-185, 2013

[6] Kozik R., Choras Michal , Flizikowski A., Theocharidou M., Rosato V., Rome E., Advanced services for critical infrastructures protection , Journal of Ambient Intelligence and Humanized Computing, vol. 6(6), 783-795, Springer, 2015.

[7] Guzmán, L., Oriol, M., Rodríguez, P., Franch, X., Jedlitschka, A., & Oivo, M. (2017, February). How Can Quality Awareness Support Rapid Software Development?–A Research Preview. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 167-173). Springer, Cham.

[8] X. Franch, C. P. Ayala, L. López, S. Martínez-Fernández, P. Rodríguez, C. Gómez, A. Jedlitschka, M. Oivo, J. Partanen, T. Raty, and V. Rytivaara: Data-driven Requirements Engineering in Agile Projects: The Q-Rapids Approach. JIT-RE 2017

[9] Apache Kafka. Project homepage: https://kafka.apache.org/

[10] Apache Sark. Project homepage: https://spark.apache.org/

[11] GitLab. Project homepage: https://about.gitlab.com/

[12] SonarQube. Project homepage: https://www.sonarqube.org

[13] A. Miller, A Hundred Days of Continuous Integration, Agile 2008 Conference, Toronto, ON, 2008, pp. 289-293.

[14] VPN, Virtual Private Network, url:
http://searchnetworking.techtarget.com/definition/virtual-private-network

[15] ACF, Autocorrelation Function, url:
https://stat.ethz.ch/R-manual/R-devel/library/stats/html/acf.html

Cluster Identification in Time Dependent Multidimensional Data

Peter Krammer[*], Marcel Kvassay[*], Ladislav Hluchý[*]

[*] Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

{peter.krammer, marcel.kvassay, ladislav.hluchy}@savba.sk

Abstract— **This paper outlines a new approach to the clustering of highly dynamic multidimensional data from agent-based simulations of crowd behaviour. It utilizes the notion of structural entropy in order to identify important periods in the genesis and evolution of clusters, and looks primarily for robust clusters comprising simulations with similar characteristics as well as developmental paths. In our previous study of these simulations we observed a surprising bifurcation of their feature-space trajectories and managed to identify its generating mechanism. The number of simulation clusters, however, remained uncertain: we tentatively assumed the existence of two global ones (simulations with timid crowd behaviour versus those that turned aggressive) but we suspected that both would resolve into further subclusters at a more detailed level of analysis. In this paper we focus specifically on the question of the number of clusters, taking into account that they can expand, contract, diverge, converge and even overlap or merge with time.**

Keywords — clustering, attribute selection, entropy, data analysis, visualization

I. Introduction

The volume of data produced by large and complex systems, be it computer simulations or real-time industrial control processes, grows incessantly. They need to be not just stored and archived, but analysed as well. The analysis is often demanding, because in these contexts it can be difficult to recognise really significant variables and periods of time even when we have their valid mathematical models at our disposal.

This article is part of a larger research programme [4], which combines structural causal analysis with machine learning methods in order to identify the key factors and mechanisms influencing the time-evolution of simulation trajectories in their feature space as well as their final outcomes. The investigated simulation scenario comes from international research project EUSAS for the European Defence Agency EDA [5] and was inspired by the recent peacekeeping ISAF mission in Afghanistan. In the scenario, a crowd of civilians is looting a shop and an approaching soldier patrol is supposed to stop the looting and disperse the crowd. The scene is depicted in Fig. 1.

The black areas represent buildings and barriers unreachable to agents. The rectangle with gray interior near the top is the looted shop. It is surrounded by dots, each representing one agent. The dark ones are the looters; the white ones are the violence-prone individuals whose intention is to attack the soldiers. The soldiers are represented by the three medium gray dots in the bottom part of the figure.

Civilian agents are endowed with one "default" motive and a matching behaviour by which they try to satisfy it. For looters this leads to looting and for the violence-prone individuals to stone-pelting the soldiers. Additionally, they possess simulated emotions of anger and fear, which too act as motives, and the corresponding aggressive and timid behaviours. Motives are restricted to the closed interval [0, 1] so that they are mutually comparable. They are evaluated periodically and the strongest wins control over the agent's behaviour.

As the patrol approaches the scene of looting, some looters may grow afraid and start leaving the scene. The violence-prone individuals, however, do not get afraid but rather attack the patrol. The violence may impact the remaining looters in two possible ways – they may either get afraid and leave, or get angry and join the attack. The ratio of looters who get afraid to those who get angry depends on their internal motivational dynamics, which we explain next.

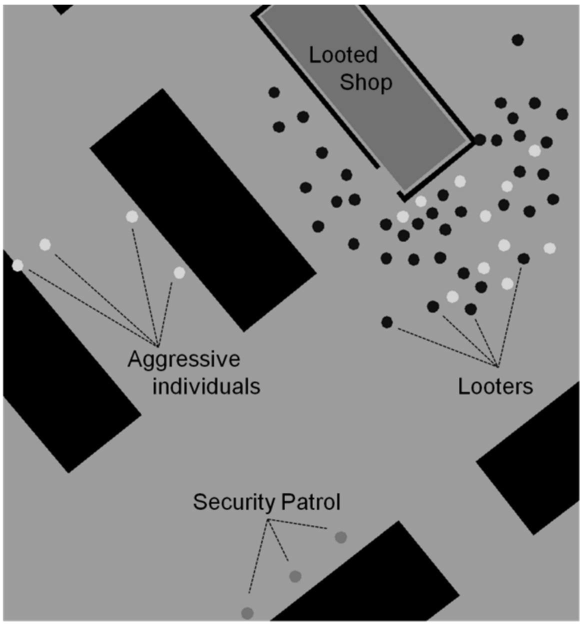


Fig. 1. The initial stage of our simulation scenario.

The dynamics of fear and anger comprise a continuous part and a discrete part linked sequentially. Their continuous dynamics is driven by the twin ordinary nonlinear differential equations (1).

$$dF/dt = c_{1F} \,.\, F \,.\, (c_{2F} - F) \,.\, (c_{3F} + I_F)$$
$$dA/dt = c_{1A} \,.\, A \,.\, (c_{2A} - A) \,.\, (I_A + L - c_{3A}) \qquad (1)$$

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

In the above equations, c_{1F}, c_{2F}, c_{3F}, c_{1A}, c_{2A}, c_{3A} are special emotion-related constants (see TABLE I.) and I_F with I_A represent the fear- and anger-related social influence of nearby agents.

TABLE I. ANGER- AND FEAR-RELATED CONSTANTS.

Name	Value	Interpretation
c_{1F}	5.0	Sensitivity of the first derivative of fear *dF/dt*
c_{2F}	1.1	Maximum value of fear *F*, at which *dF/dt* becomes zero
c_{3F}	0.1	Inbuilt tendency of fear to increase
c_{5F}	1.0	Sensitivity to fear-inducing events
c_{1A}	2.0	Sensitivity of the first derivative of anger *dA/dt*
c_{2A}	1.1	Maximum value of anger *A*, at which *dA/dt* becomes zero
c_{3A}	0.3	Resilience to anger
c_{5A}	1.0	Sensitivity to anger-inducing events

During simulation, these differential equations are solved numerically. Thus, at each simulation step, new preliminary "continuous" values of both fear and anger $F(t + \Delta t)$, $A(t + \Delta t)$ are calculated from their "previous" total values $F_T(t)$, $A_T(t)$. Then the discrete dynamics come into play, driven by the twin difference equations (2).

$$F_T(t + \Delta t) = F(t + \Delta t) + c_{5F} \,.\, \Delta E_F$$
$$A_T(t + \Delta t) = A(t + \Delta t) + c_{5A} \,.\, \Delta E_A \qquad (2)$$

In the above equations, c_{5F} and c_{5A} are sensitivity constants (see TABLE I.) and ΔE_F with ΔE_A represent the cumulative impact of the external events perceived by the agent during the simulation step under consideration. Main event impacts on fear and anger are listed in TABLE II.

In the next iteration these new total values of fear F_T and anger A_T will be used as initial conditions in the numerical solution of the differential equations (1) representing their continuous dynamics. This sequential coupling of the continuous and the discrete dynamics qualifies our agent model as *sequential hybrid* in the sense of Swinerd and McNaught [14]. Further details about the model and its equations can be found in [4].

TABLE II. MAIN EVENT IMPACTS ON FEAR AND ANGER.

	Impact on Fear		Impact on Anger	
Event	Direct	Indirect	Direct	Indirect
Effective shot	0.400	0.350	0.100	0.250
Warning shot	0.300	0.300	0.100	0.100
Stone thrown	0.002	0.002	0.180	0.150

In our previous work we focussed primarily on the modelling and prediction of the final outcome of this scenario [6, 8]. It could be broadly characterised as a "security success" or a "security failure" depending on how many civilians got angry and attacked the security patrol. This outcome could be gleaned from various statistics recorded during the simulation, such as the number of stones thrown or gun shots fired.

In order to successfully predict and explain the final outcome of our simulations, we needed to identify, first, their crucial moment, i.e. the earliest point in simulated time at which their final outcome could be reliably predicted, and, second, the key attributes carrying that information. In fact, there were only two possible final outcomes: the crowd behaviour in each simulation could turn either predominantly aggressive ("security failure") or predominantly timid ("security success"). This roughly corresponded to the two global clusters clearly visible in the two-dimensional attribute space (F-Count, A-Count) shown in Fig. 2 on the basis of data from the 90th second of simulated time. The attributes A-Count and F-Count stand for how many times during the simulation anger or fear, respectively, prevailed in some civilian agent and dominated its external behaviour.

We are now shifting our focus to how this scenario unfolds in time in order to find out how many clusters of simulations with similar characteristics and developmental paths are really there. To this end we shall use not only the final attribute values from the 90th second, but their previous recorded values as well.

We expect that these clusters should occupy relatively compact and continuous areas of the feature or attribute space. Ideally, each cluster should be separated from others by some empty or sparsely-populated margin, although we cannot rule out that some might be contiguous or even overlapping. Regarding their time-evolution, the simulations in the same cluster should have similar developmental paths, i.e. we expect the clusters (once they are clearly formed) to visibly "travel" through the attribute space, possibly further expanding, contracting or even colliding with each other along the way.

We believe this kind of analysis can significantly deepen our understanding of the processes and mechanisms driving the simulations. If, for example, there are three distinct directions that each simulation can take, it might be advantageous to model each direction separately since it is quite likely that each is driven by its own specific mechanisms and forces. Moreover, each might have its own set of key attributes that determine the unfolding of simulations along its particular path.

The ability to distinguish important cases could also lead to considerably simpler analysis and modelling: the models could then be case-specific, each focussing only on the attributes and mechanisms important for one particular case. Such an approach, however, requires a highly representative set of data.

In our analysis we shall reuse the target attribute and the classification model from our previous work [4]. Both were calculated from the attributes A-Count and F-Count, which stand for how many times during the simulation anger or fear, respectively, prevailed in some civilian agent and dominated its external behaviour. We used SVM (Support Vector Machine) to train a classification model from the set of A-Count and F-Count values recorded at the 90th second of simulated time. This model was defined by formula (3), which assigned each simulation to one of the two global clusters: a positive value meant that the crowd behaviour in the simulation remained largely timid, while a negative one signalled aggressive developments.

$$g = -0{,}3846\,.\,\text{A-Count} + \\ + 0{,}0769\,.\,\text{F-Count} + 4{,}2308$$

$$cluster = \begin{cases} 1 \; if \; g \geq 0 \\ -1 \; if \; g < 0 \end{cases} \tag{3}$$

From this model, the decision boundary was obtained by setting g equal to zero, which allowed us to express the boundary line by the equation (4):

$$F\text{-}Count = 5{,}0013\ .\ A\text{-}Count - 55{,}0169 \tag{4}$$

This model was originally derived from an early batch of 350 simulations. At present we have at our disposal a much larger dataset with 1500 simulations. Sixteen of these were excluded from further analysis because their data exhibited significant anomalies. For the remaining 1484 simulations we now need to check whether the model (3) can still correctly assign them to the two global clusters. As we can see in Fig. 2, the decision line given by function (4) still neatly separates the two global clusters with a considerable margin. It means that we can safely proceed to inquire into whether or not they might be composed of smaller subclusters.

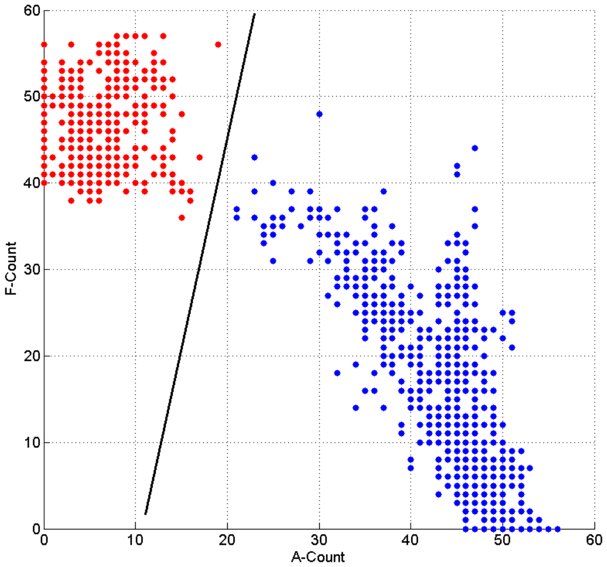


Fig. 2. A visual check of how adequately the decision boundary given by equation (4) separates the two global simulation clusters at the 90th second of simulated time. The first (red) global cluster in the top left corner is characterised by low numbers of angry agents (A-Count) and can be considered a “security success”. The second (blue) global cluster represents “security failures” with high numbers of angry agents.

II. Data Reduction

Fine-grained analysis of dynamic multidimensional data is often a challenge, especially for large datasets. One of the reasons is that standard visualisation methods used in the analysis can meaningfully display only a limited number of dimensions. Some of the more advanced ones [1, 2] can display more dimensions, but they too suffer from certain limitations. Often they are not universal and work only in some domains and with some data types or cannot visualise certain specific aspects of data.

It is therefore advisable to reduce the number of attributes (and, consequently, of dimensions) by using only the most significant ones. There are several criteria and algorithms for determining attribute significance (or attribute selection [9]), such as Entropy, Information Gain [10], Information Gain Ratio [10], Gini Index [10], Chi^2 [3], Correlation Coefficient, Relief algorithm [11, 12], etc. [13].

In our case, since the input attributes were numeric, the most suitable criterion turned out to be Chi^2. We applied it separately to each subset of our data, which consisted of attribute values recorded at the same moment of time. Each simulation covered 90 seconds of simulated time and the attribute values were recorded periodically every two seconds. In this way we could gauge the time evolution of significance for each input attribute. Attribute names and meanings are listed in TABLE III.

TABLE III. List of attributes and their meanings.

Name	Interpretation
Avg-Anger	Value of anger averaged over all civilian agents participating in the scenario
Avg-Fear	Value of fear averaged over all civilian agents in the scenario
A-Arousal	Portion of Avg-Anger attributed to the causal influence of the model variable arousal L
A-ExtC	Portion of Avg-Anger attributed to the causal influence of actions of civilian agents
A-ExtS	Portion of Avg-Anger attributed to the causal influence of actions of security patrol
A-Init	Portion of Avg-Anger attributed to the setting of initial value of anger
A-Sat	Portion of Avg-Anger attributed to the mechanism restricting the value of anger to interval [0, 1]
A-Soc	Portion of Avg-Anger attributed to social influence of nearby agents I_A
F-ExtC	Portion of Avg- Fear attributed to the causal influence of actions of civilian agents
F-ExtS	Portion of Avg- Fear attributed to the causal influence of actions of security patrol
F-Init	Portion of Avg- Fear attributed to the setting of initial value of fear
F-Soc	Portion of Avg- Fear attributed to social influence of nearby agents I_F
MoE-eff-shots	Number of effective gun shots by the security patrol
MoE-warn-shots	Number of warning gun shots by the security patrol
Moe-stones-thrown	Number of stones thrown at the security patrol

TABLE IV. and TABLE V. show the significance for these input attributes as evaluated by the Chi^2 criterion. We are only showing selected points of simulated time, with finer granularity in the interval between the 10th and the 30th second, which turned out to be the most important and dynamic regarding the time-evolution of attribute significance as well as prediction accuracy.

The attributes F-Count and A-Count were omitted from these tables since they were used to derive the target attribute. The attribute F-Sat was also omitted because its actual value was nearly always zero and, consequently, its significance level was nil at all the recorded points.

TABLE IV. COMPARISON OF ATTRIBUTE SIGNIFICANCE CALCULATED BY THE CHI^2 CRITERION, PART 1.

simulation time	Attribute							
	Avg-Anger	Avg-Fear	A-Arousal	A-ExtC	A-ExtS	A-Init	A-Sat	A-Soc
10	298	38	0	303	38	0	0	34
12	631	815	97	611	804	111	8	567
14	716	1149	66	834	1001	394	90	719
16	772	1235	110	719	1053	563	93	846
18	810	1296	151	623	1107	656	93	846
20	858	1347	211	533	1130	735	92	939
24	914	1395	344	506	1160	858	91	954
30	1000	1416	439	519	1154	971	88	945
40	1196	1410	461	449	1125	1094	93	982
50	1407	1401	418	410	1094	1207	92	1017
60	1468	1378	417	486	1086	1293	93	1034
70	1474	1345	316	552	1077	1356	94	1015
80	1465	1341	206	533	1087	1398	89	980
90	1465	1340	153	519	1100	1417	90	927

TABLE V. COMPARISON OF ATTRIBUTE SIGNIFICANCE CALCULATED BY THE CHI^2 CRITERION, PART 2.

simulation time	Attribute						
	F-ExtC	F-ExtS	F-Init	F-Soc	MoE-eff-shots	Moe-warn-shots	Moe-stones-thrown
10	309	38	29	57	0	55	369
12	606	813	329	933	0	239	145
14	945	891	1201	1296	0	28	44
16	991	716	1346	1333	0	24	0
18	1032	561	1380	1359	0	33	0
20	1062	360	1404	1368	0	21	305
24	1093	133	1424	1378	0	291	673
30	1148	116	1441	1379	620	507	858
40	1175	198	1449	1376	938	874	1075
50	1187	300	1462	1369	1110	1021	1172
60	1182	406	1464	1360	1183	1140	1222
70	1178	520	1461	1357	1312	1231	1323
80	1179	601	1461	1344	1372	1243	1345
90	1181	679	1464	1342	1377	1249	1357

Next, the overall significance of each attribute was determined as the average of its significance estimates at all the recorded points (i.e. not only those shown in TABLE IV. and TABLE V.). We then chose eight attributes with the highest overall significance for further in-depth analysis: F-Soc, F-Init, Avg-Fear, Avg-Anger, F-ExtC, A-ExtS, A-Init, A-Soc.

Our next sub-goal was to identify the most important period in the evolution of simulations that determined their subsequent development. We expected this would be the right moment to look for robust clusters of simulations with similar characteristics and developmental paths.

To this end, we employed the criterion of entropy. We first calculated it from the eight most important attributes listed above and then repeated the procedure after adding two attributes from which our target class was derived (A-Count and F-Count). In the process, we divided the min-max scope of each attribute into seven equal intervals. For ten attributes this divided the occupied portion of the attribute space into 7^{10} equal hypercubes. For each hypercube we then calculated the number of simulations that it contained and from these counts we calculated the entropy. For two attributes (and, consequently, a two-dimensional attribute space), the entropy can be calculated by formula (5). For eight- or ten-dimensional attribute space the formula would be analogical but would involve eight or ten nested sums, respectively. Of course, the min-max scope of each attribute could also be divided into more than seven intervals: this was a practical decision that reflected a working compromise between the number of attributes and the available computing power.

$$E(t) = \sum_{j=1}^{7} \sum_{k=1}^{7} \frac{n_{j,k}(t)}{N} . log_2 \left(\frac{n_{j,k}(t)}{N} \right) \qquad (5)$$

In the above formula, E(t) stands for the entropy of the occupied portion of a hypothetical two-dimensional attribute space at time t, while N represents the number of simulations (i.e. samples). Since all the simulations run through all the recording points, N does not depend on time t. Indexes j and k denote a particular square (two-dimensional hypercube) in the occupied portion of the attribute space and $n_{j,k}(t)$ represents the number of simulations residing in that square.

Before actually calculating the entropy, we decided to normalize the attribute values. In this respect, there were two options open to us:

1. We might treat all the recorded attribute values as one huge dataset regardless of when they were recorded, find the global minimum and maximum for each attribute and use these to normalize the data. We shall call the entropy calculated in this way the “global entropy”.
2. We might also normalize the attributes for each recorded point in time separately by using only the “temporal” minima and maxima that actually occurred until that moment. We shall call this kind of entropy “structural” because it better reflects the internal or structural complexity of our clusters as defined by their shape and relative position.

We preferred the second option because it offered a way to suppress the effect of the general tendency of our attributes to grow with time. We then calculated the structural entropy for all the recording points and we show its time-evolution for the ten most significant attributes in Fig. 3, and for the eight most significant ones in Fig. 4.

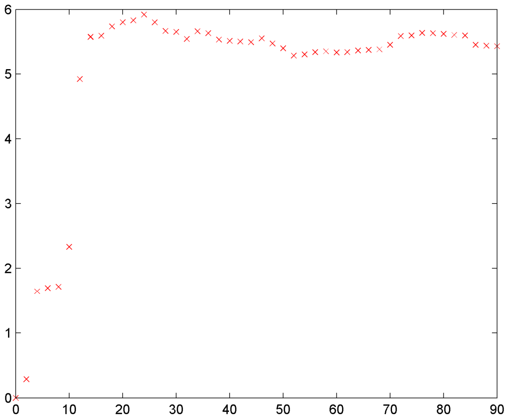


Fig. 3. The time evolution of structural entropy (vertical axis) calculated from the ten most significant attributes. Horizontal axis shows simulated time in seconds.

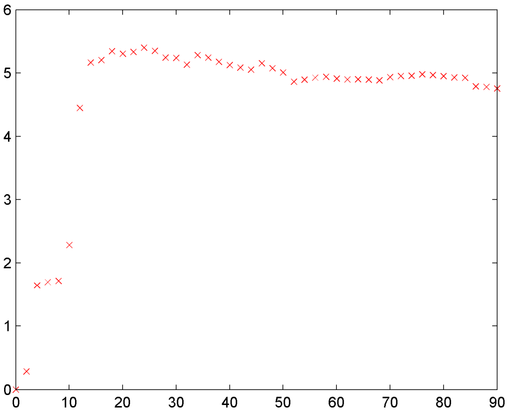


Fig. 4. The time evolution of structural entropy (vertical axis) calculated from the eight most significant attributes. Horizontal axis shows simulated time in seconds.

The two graphs are quite similar. Here are some of their salient features:

- the structural entropy of our simulation clusters grew most quickly between the 10th and the 12th second of simulated time
- At 14 seconds we noticed a kind of "saturation" and the structural entropy stabilized soon after it crossed the value of 5. Its subsequent minor fluctuations did not appear significant.
- At 24 seconds, the structural entropy reached its global maximum, although its actual increase from its previous level at 22 seconds was rather small.

For comparison, Fig. 5 shows the time evolution of "global entropy" calculated from the ten most significant attributes. Its continuous growth almost till the end of the simulation effectively obscures the importance of the early period between the 10th and the 24th second, which contained all the interesting features: the moment of the fastest growth, the global maximum, and the beginning of saturation. Our further analysis therefore focused primarily on this early decisive period. We present its results in the next section.

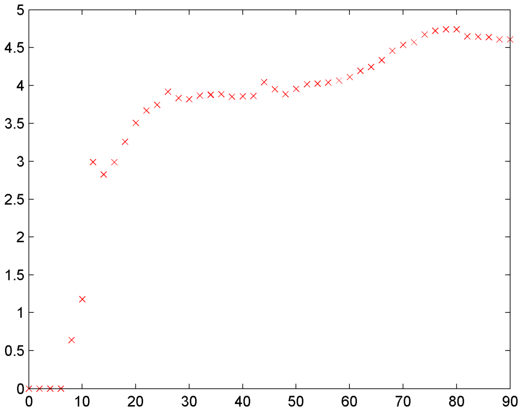


Fig. 5. The time evolution of global entropy (vertical axis) calculated from the ten most significant attributes. Horizontal axis shows simulated time in seconds.

III. Data Analysis

To find groups of similar simulations in dynamic data by direct application of standard clustering algorithms can be problematic: we do not know in advance their number, shape, mutual distances or sizes. Moreover, at some points of simulated time they may not be clearly distinguishable and may even melt into each other without any visible boundaries separating them. For these reasons we decided to identify the new subcluster structure manually on the basis of visualizations of our reduced set of data.

Each visualisation consisted of a time series of batches of two-dimensional charts in the attribute space. Each batch was constructed from the data recorded at the same moment of time. Each batch thus represented a snapshot of simulation positions in the attribute space at that moment. Individual simulations were represented by dots and their cluster assignment was shown by the dot's colour. We used such snapshots for all the recorded times (from the 2nd till the 90th second), although we focussed primarily on the period between the 10th and the 24th second, in which most of the interesting developments were found to occur. Our visualisations were in fact a kind of animation showing the same arrangement of charts at different points of simulated time.

As an example, in Fig. 6 we show the snapshot from the 14th second. If we consider only the spatial distribution of the dots in the four charts in this figure, there appear to be several areas of increased density which could be regarded as candidate subclusters. They are most clearly discernible in the top left chart, i.e. in the two-dimensional attribute space (Avg-Anger, Avg-Fear). At the same time, their colour coding corresponds to how they later combine into the two global clusters at the 90th second of simulated time shown in Fig. 2: the relatively compact group of red dots in Fig. 6 remains compact and forms the first (red) global cluster in the upper left corner of Fig.2, while the other candidate subclusters in Fig. 6 gradually coalesce into the second (blue) global cluster protruding from the lower right corner of Fig.2 towards its centre. Overall, then, Fig. 6 seems to support our original intuition that at least the second global cluster could be considered as having evolved from several smaller subclusters.

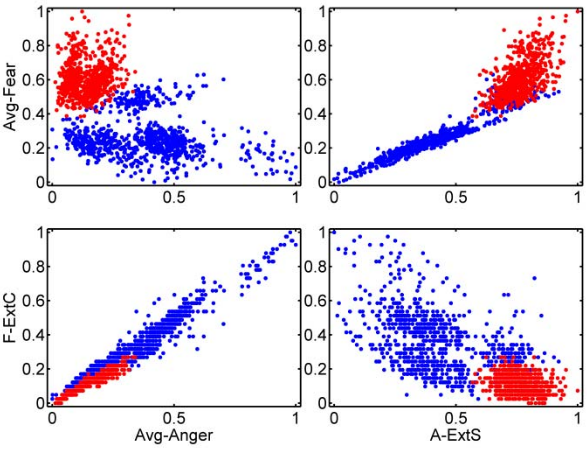


Fig. 6. Scatter plots of emerging simulation subclusters at the 14th second of simulated time. Each dot represents one simulation and its colour shows to which of the two global clusters it will be ultimately assigned due to its final position at the 90th second of simulated time.

In order to establish the number of subclusters present in Fig. 6 with more rigour, we relied primarily on the data from the period between the 10th and the 24th second of simulated time, which was identified as significant by our criterion of structural entropy. The most informative moment for the identification of new subclusters actually turned out to be the 14th second.

We first focused on subcluster boundaries, which we determined iteratively. We started by identifying new subcluster candidates through visible margins separating them from other candidates on at least some charts in our snapshot animations. These margins then defined their provisional boundaries. Next, these candidates were assigned different colours and the animation was re-run to see whether they stayed distinct and reasonably compact under different views (i.e. when plotted against different pairs of input attributes) during a significant portion of the simulation. We did not require them to stay distinct till the end of the simulation because we in fact expected them to gradually coalesce and form the two global clusters shown in Fig.2. As this process was iteratively repeated, their tentative boundaries were fine-tuned and ultimately fixed in their final positions visualised in Fig. 7 through colour coding. Fig. 7 really shows the same data from the 14th second as Fig. 6, but partitioned into four subclusters. We could of course produce many more charts like this for other combinations of input attributes but space limitations prevent us from including them here. We visually checked a considerable number and quite a few showed the four new subclusters as reasonably compact and distinct from each other.

We next displayed these four new subclusters at all the recorded points of simulated time (2nd to 90th second) in order to check to what extent they would overlap and mix. Of course, the most interesting for us was the final, 90th second which showed the cumulative effect of all the changes since the 12th second when these subclusters first visibly emerged. An assortment of the views of these four subclusters at the 90th second for different pairs of input attributes is shown in Fig. 8 and 9.

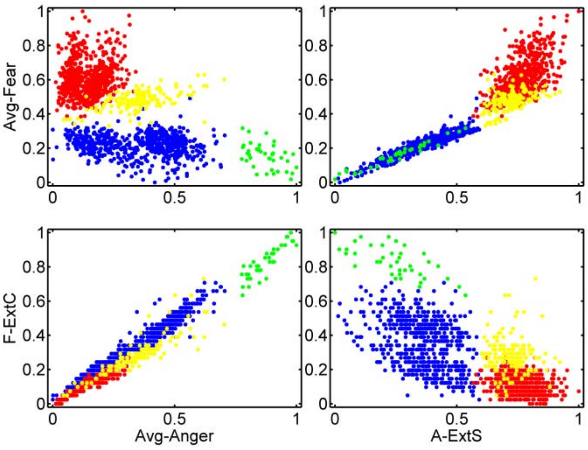


Fig. 7. Visual representations of four new clusters in the 14th second of simulated time.

We can see that in Fig. 8 and 9 some of the subclusters have become contiguous, i.e. the empty spaces separating them have vanished, yet there is only a minimal overlap and mixing among them. It is evident that the colours are not distributed randomly, but rather fill distinct and reasonably compact areas of the attribute space. Moreover, the yellow subcluster, which in Fig. 7 appeared to be closely related with the red one, has grown closely associated with the blue subcluster in Fig. 8 and 9. That is why it is important to investigate the time evolution of subclusters: at the moment when they are most clearly discernible (in this case, at the 14th second of simulated time), their ultimate affiliation may not yet be apparent.

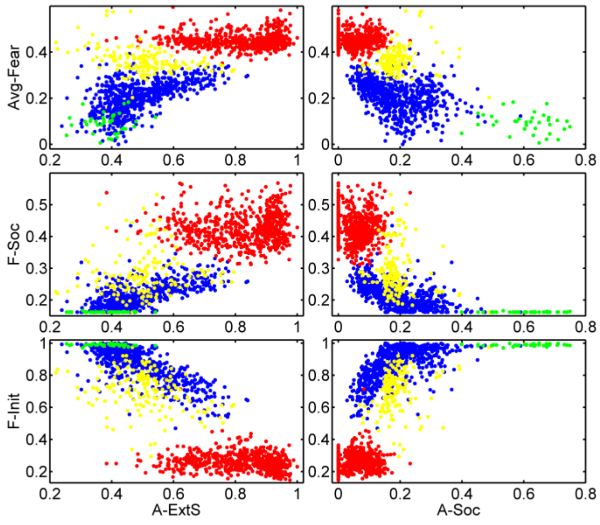


Fig. 8. Visual representations of the four detected classes in the 90th second of simulated time – Part I.

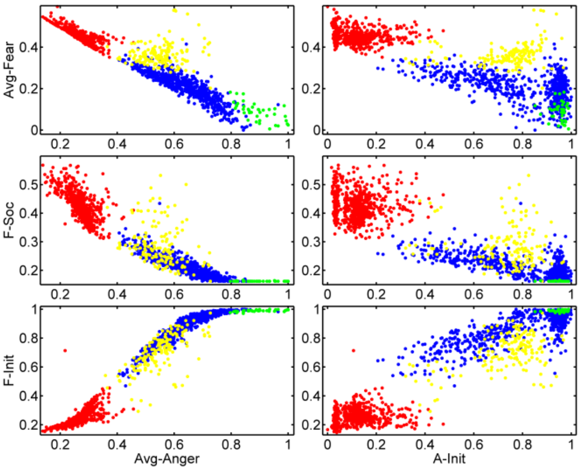


Fig. 9. Visual representations of the four detected classes in the 90th second of simulated time – Part II.

In order to validate our results independently, we again relied on the data from the 14th second. We applied to it selected clustering methods compatible with the character of our domain, namely Weka's implementations of Kohonen's Self-Organising Map (SOM) and Hierarchical Clusterer with links of Ward type. The main factors qualifying these methods were their preference for compact cluster shapes and their ability to take into account the width of margins separating the clusters. We started in two-dimensional attribute space (Avg-Fear, Avg-Anger) which our domain knowledge indicated to be the most relevant for the emergence and further evolution of subclusters.

We first applied SOM with 1000 convergence epochs, 2000 ordering epochs, lattice height = 2, lattice width = 2, and learning rate = 1.0. For these parameter values, SOM identified four clusters shown in Fig. 10, whose shapes were found to be quite robust with respect to minor variations of SOM parameter values.

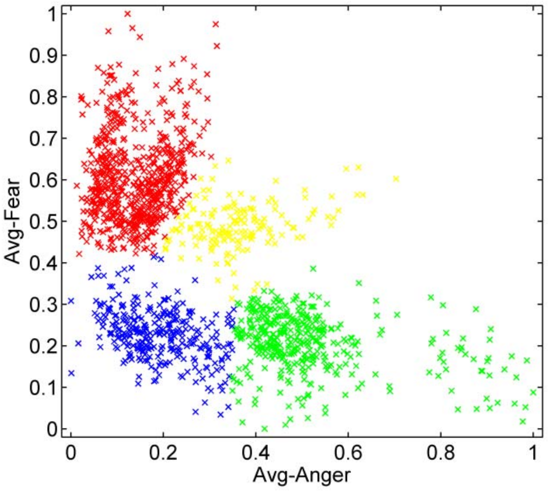


Fig. 10. SOM clusters identified on the basis of two attributes Avg-Anger, Avg-Fear and their values at the 14th second.

We can see that the clusters identified by SOM in Fig. 10 are very similar to those that we have identified manually in the top-left frame of Fig. 7. The only significant discrepancy is the different position of the border separating the green cluster from the blue one in the lower portion of both charts. It is a known fact that humans intuitively tend to place such borders in the widest margins (effectively ignoring the unbalance in the size of the resulting clusters), while clustering algorithms using variance and similar criteria like squared error or Euclidean distance tend to factor in cluster sizes as well. Therefore, despite this minor difference, we consider this to be an excellent fit. Moreover, nearly identical clusters as those in Fig. 10 were obtained through the Weka's hierarchical clusterer using Ward links and Euclidean distance, when we forced it to search for four clusters.

As we have already mentioned, in this first validation phase the clustering algorithms operated on a reduced set of two-dimensional attribute data (Avg-Fear, Avg-Anger) from the 14th second, because there the shapes of the emerging clusters were most clearly visible. In the second validation phase we still used only the data from the 14th second, but we now supplied our clusterers with all ten significant attributes identified in Section II. When we ran the Weka's hierarchical clusterer with Ward links and Eclidean distance on this data and forced it to search for five clusters, we obtained the clusters shown in Fig. 11.

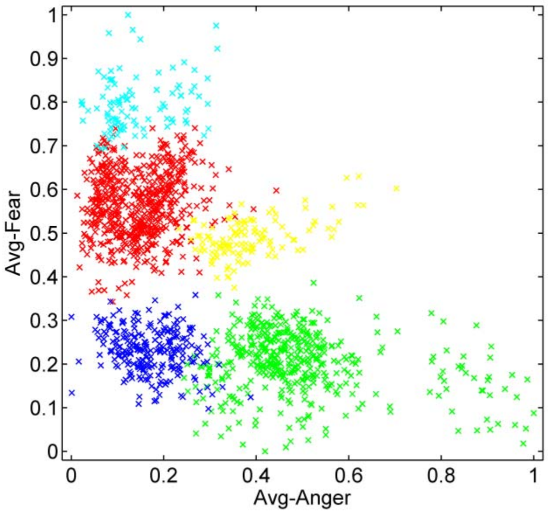


Fig. 11. Hierarchical clusters identified on the basis of ten most significant attributes and their values at the 14th second.

We can see that their shapes still correspond very closely to those in Fig. 10, the only major exception being an additional cyan-coloured cluster in the top-left corner of the chart, which was split off the red cluster. The Weka's hierarchical clusterer considered this cyan cluster as more different from the rest of the red cluster than the yellow cluster, i.e. when forced to search for four clusters only, it merged the yellow cluster with the red one and kept the cyan one as the fourth. Nevertheless, it is still a very good fit, the discrepancy again stemming mainly from the difference between the human intuitive way of splitting clusters and that of computer algorithms minimizing variance and similar criteria. Moreover, we have obtained a very similar cluster structure as that shown in Fig. 11 through Kohonen's Self-Organizing Map for 1000 convergence epochs, 2000 ordering epochs, lattice height = 3, lattice width = 2, and learning rate = 1.0. On this basis we feel entitled to claim that the "objective" existence of the four subclusters shown in Fig. 7 and 8 was indeed established.

As for the eventual practical impact of our conclusions in the context of our simulation scenario, it would depend on the purpose with which the intended users of the EUSAS system (security analysts and trainers) would conduct these simulations. In all likelihood they would try to verify the effectiveness and robustness of the tactics used by the simulated security patrol in a representative subset of admissible simulation settings. In other words, they would try to see whether the strategy was worthwhile and stable enough before they would try to implement it in real life. Should it be found deficient in some respects, they might want to know the reasons and how it could be improved. Towards this end, they would probably start by inspecting some general statistics, or measures of effectiveness (MoE), characterising the overall results of each simulation run, i.e. how well the security patrol managed the situation. Typical MoE used for the purpose are the numbers of effective and warning shots and of thrown stones, listed in the last three rows of TABLE III. Each simulation run in the batch could then be described by these three MoE values and represented as one "data-point" in the corresponding three-dimensional attribute space. A three-dimensional structure, similar in character to the two-dimensional one in Fig. 2, would emerge, in which successful handling of the scenario would be signalled by low values of these MoE, and failure by their high values.

Seeing that more than half of the simulations exhibited signs of "security failure" (specifically those in the second, i.e. blue global cluster in Fig.2), the analysts and trainers would probably want to know the reasons. In this particular case, high numbers of gun shots by the security patrol in the "failing" scenarios were a legitimate response to intense stone-pelting, which in turn stemmed from high numbers of angry civilians. At this point it would make sense to inquire into how many agents grew angry or afraid during each simulation, which would lead directly to the chart in Fig. 2.

A deeper inquiry into security failures would then lead the analysts to the detailed examination of the time-evolution of fear and anger in civilian agents in search for their root causes. The full explanation, into which we cannot go here, would involve intricate details of causal partitioning of fear and anger that we described in [4]. Restricting ourselves to a brief indication, we might say that each simulation run of a given scenario – whether successful or not – can be characterised by the sequence of civilian and security actions that occurred in it. In other words, there may be several ways to succeed, some better than others, and several ways to fail, some worse than others. It would be impractical to record and analyse the complete action sequences in detail, but we discovered in [4] that important aspects (including information with high predictive power) were preserved in our causal attributes and in the time-evolution of clusters and subclusters in their feature space. As a result, important insights into root causes of security failures might be gleaned from how the two global clusters in Fig. 2 emerge from smaller subclusters in Fig. 7, because each subcluster is likely to represent simulations with similar action sequences. A deeper analysis of these sequences, which we intend to do in our future work, should then lead us to the root causes of both security successes and security failures.

IV. Conclusion and Future Work

We have analysed highly dynamic multidimensional data and managed to identify in them a group of four simulation subclusters that preserved their separate identity and relatively compact shape since the moment of their emergence (Fig. 7) practically till the end of the simulation (Fig. 8 and 9). Their stability and robustness was independently validated by two clustering algorithms, the Weka's hierarchical clusterer and Kohonen's self-organising map, in the attribute data from the 14th second of simulated time. In the two-dimensional attribute space (Avg-Anger, Avg-Fear) the fit was nearly perfect, except that one border was placed differently (cf. Fig. 10 and the top-left frame of Fig. 7). In the ten-dimensional attribute space there appeared one more discrepancy, because one of our manually identified subclusters (the red one) was further split by both clusterers (cf. Fig. 11 and the top-left frame of Fig. 7). Nevertheless, the overall fit remained very good. We can therefore conclude that all the four subclusters identified in Fig. 7 are not only robust and stable, but also consist of simulations with similar characteristics and action sequences.

In the data reduction phase of our analysis we have relied on the Chi^2 criterion and a peculiar variety of information entropy that we termed "structural entropy". This is entropy calculated from the data recorded at one particular moment in the evolution of clusters and normalised by the minimum and maximum attribute values that actually occurred until that point. These "temporal" minima and maxima define at each moment what we could call an "observable universe", that is, a subset of attribute space actually occupied by the simulations. This dynamic scaling turned out to be crucial for suppressing the effect of our continuously expanding "observable universe" and for identifying the time period in which most of the significant developments occurred (from the 10th till the 24th second of simulated time). Had we scaled our data globally on the basis of global attribute minima and maxima, we might have easily missed the early onset of the internal complexity of our clusters, because in the beginning they all occupied a relatively small portion of the "global universe", and so their global entropy might appear small regardless of their complex internal structure, as demonstrated in Fig. 5. On the other hand, Fig. 3 and 4 proved that this complex internal structure emerged very rapidly and was clearly in place by the 14th second of simulated time. Equally gratifying for us was the fact that the start of the "important" period (indentified here through domain-independent means, i.e. structural entropy) neatly coincided with the key moment deciding the future course of these simulations identified in our previous work [4] by relying on a rather deep domain-specific knowledge. We therefore feel that structural entropy and related general techniques represent a promising approach for the analysis of large and dynamic multidimensional data.

In the future we plan to combine such general techniques with domain-specific knowledge so as to analyse our simulation subclusters in a yet greater detail. In particular, we expect that the mechanism responsible for the bifurcation of simulation paths identified in [4] could shed further light on their composition and evolution, as well as on the root causes of the observed security successes and failures.

ACKNOWLEDGMENT

This work was supported by the VEGA project 'Methods and algorithms for the semantic processing of Big Data in distributed computing environment' (2/0167/16).

REFERENCES

[1] Robert Haralick: Handbook of Data Visualization, ISBN 978-3-540-33036-3, Springer 2008, accessed 12.9.2017: http://haralick.org/DV/Handbook_of_Data_Visualization.pdf

[2] Principles of Data Visualization, accessed 12.9.2017: http://www.fusioncharts.com/whitepapers/downloads/Principles-of-Data-Visualization.pdf

[3] George Forman: An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research 2003, p. 1289-1305, accessed 12.9.2017: http://www.jmlr.org/papers/volume3/forman03a/forman03a_full.pdf

[4] M. Kvassay, P. Krammer, L. Hluchý, B. Schneider, "Causal Analysis of an Agent-Based Model of Human Behaviour," *Complexity*, vol. 2017, Article ID 8381954, 18 pages, 2017. doi:10.1155/2017/8381954

[5] M. Kvassay, L. Hluchý, Š. Dlugolinský, B. Schneider, H. Bracker, A. Tavčar, M. Gams, M. Contat, Ł. Dutka, D. Król, M. Wrzeszcz, J. Kitowski. A novel way of using simulations to support urban security operations. In *Computing and Informatics*, 2015, vol. 34, no. 6, p. 1201-1233. ISSN 1335-9150.

[6] P. Krammer, M. Kvassay, L. Hluchý: Validation of Parameter Importance by Regression, in INES 2014, pp 27 - 32, ISBN 978-1-4799-4616-7.

[7] J. M. Kvassay, L. Hluchý, P. Krammer, B. Schneider: Causal Analysis of the Emergent Behavior of a hybrid Dynamical System, Acta Polytechnica Hungarica 2014, Vol. 11, num. 4, pp. 21-40, ISSN 1785-8860.

[8] M. Kvassay, P. Krammer, L. Hluchý: Validation of Parameter Importance in Data Mining: A Case Study, in WIKT 2013, pp. 115 - 120, ISBN 978-80-8143-128-9, accessed 12.9.2017: http://web.tuke.sk/fei-cit/wikt2013/wikt%202013.pdf

[9] Isabelle Guyon, André Elisseeff: An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3 (2003), p. 1157-1182, accessed 12.9.2017: http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[10] R.C. Barros et al.: Automatic Design of Decision-Tree Induction Algorithms, Springer Briefs in Computer Science 2015

[11] Wout Megchelenbrink: Relief-based feature selection in bioinformatics, 2010.

[12] R. P. L. Durgaba: Feature Selection using ReliefF Algorithm, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, 2014, ISSN 2278-1021.

[13] Jasmina Novakovic, Perica Strbac, Dusan Bulatovi: Toward optimal feature selection using ranking methods and classification algorithm, Yugoslav Journal of Operations Research 2011, accessed 12.9.2017: http://elib.mi.sanu.ac.rs/files/journals/yjor/41/yujorn41p119-135.pdf

[14] C. Swinerd, K. R. McNaught, "Design classes for hybrid simulations involving agent-based and system dynamics models," *Simulation Modelling Practice and Theory*, vol.25, pp. 118–133, 2012.

Vertebrae detection in X-ray images based on deep convolutional neural networks

Kurachka K. S., Tsalka I.M.
Pavel Sukhoi State Technical University of Gomel
Gomel, Republic of Belarus
Email: kurochka@gstu.by, tsobako@gstu.by

***Abstract*—We proposed a model for automated vertebrae detection in X-ray image which can be used in osteoporosis diagnosis. We used deep convolutional neural network as classification method. During the experiments we found out that the model satisfies the requirements and is acceptable for problems of this type.**

***Keywords*—digital image processing, X-ray images, spine images, convolutional neural networks**

I. Introduction

Osteoporosis is one of the most common diseases which lead to disability. Main approaches which allow to diagnose osteoporosis without surgical intervention are computer tomography, magnetic resonance imaging and radiography. First two approaches are often used for vertebrae and intervertebral disk state study and three-dimensional models reconstruction [1]. But on the beginning of the illness can be difficult to diagnose patient only with vertebrae state visual analysis. There is used more precise methods based on vertebrae's size and relative position [2].

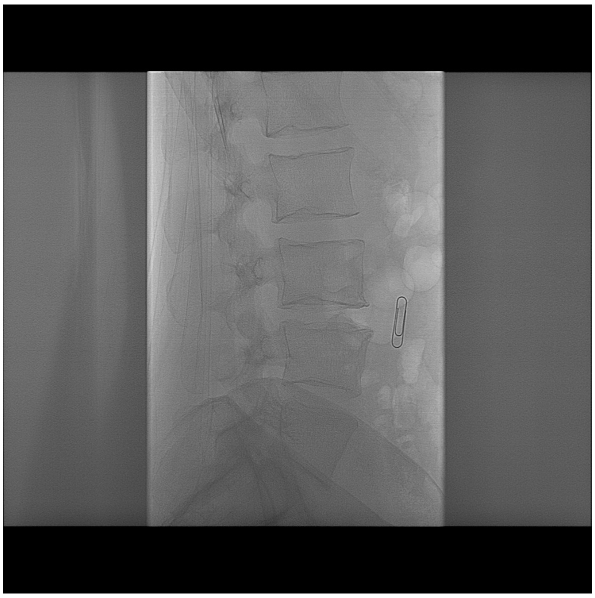


Figure 1. Vertebrae X-ray image example

Spine study are performed at Gomel clinical regional hospital for diagnosing and treating osteoporosis in Republic Belarus. Philips Allura XP FD2 X-ray radiograph is most used device to studying. Physicians use the device to get side images of spine and determine following geometrical parameters [2]:

- Sizes and inclination angles of the vertebrae
- Sizes and angles of intervertebral discs

The main disadvantage of this process is the accuracy of determining parameters fully depends on physician professional skills and diagnostic results are fully subjective.

We offer to make automatic vertebrae detection in images to speed up and improve accuracy of diagnosis.

II. Problem formulation

We suppose X-ray images are two-dimensional functions $\omega(x, y) : D_\omega \to \mathbf{Q}$, that maps coordinates $(x, y) \in D_\omega \subseteq \mathbf{Z}^2$ to x-ray density at the coordinate. Here is $\mathbf{Z}$ - set of integer numbers and $\mathbf{Q}$ - set of density levels $\mathbf{Q} = [0; Q - 1] \cap Z$.

Let $\mathbf{\Omega}$ – set of studied images. Set contains images $g_1, g_2, \ldots, g_n$ with the following conditions $g_1 \cap g_2 \cap \ldots \cap g_n = \emptyset$ and $g_1 \cup g_2 \cup \ldots \cup g_n = \mathbf{\Omega}$.

Solving the main problem is determing function $C(\omega) : \mathbf{\Omega} \to P$, which predicts probability $P = [0, 1]$ of vertebrae existence on image and we necessarily to need all the possible images for function determining.

In this work we use deep convolutional neural network as the classification function.

III. Previous works

Multiple object localization on image is one of most the important tasks. A lot of works were dedicated, in particular, medical images.

Authors of the work [3] suggest fast spine detection method on medical image. But the algorithm doesn't localize separate vertebrae and its primary target is scoliosis study.

Work [4] describes an approach with using SIFT–descriptors and training are based on support vector machines (SVM). But SIFT works not accurate enough for our images.

Authors of the work [5] suggest vertebrae centroids detection, but for the diseases of the musculoskeletal system diagnosis it's needed to determine exact location and geometric sizes of vertebrae.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

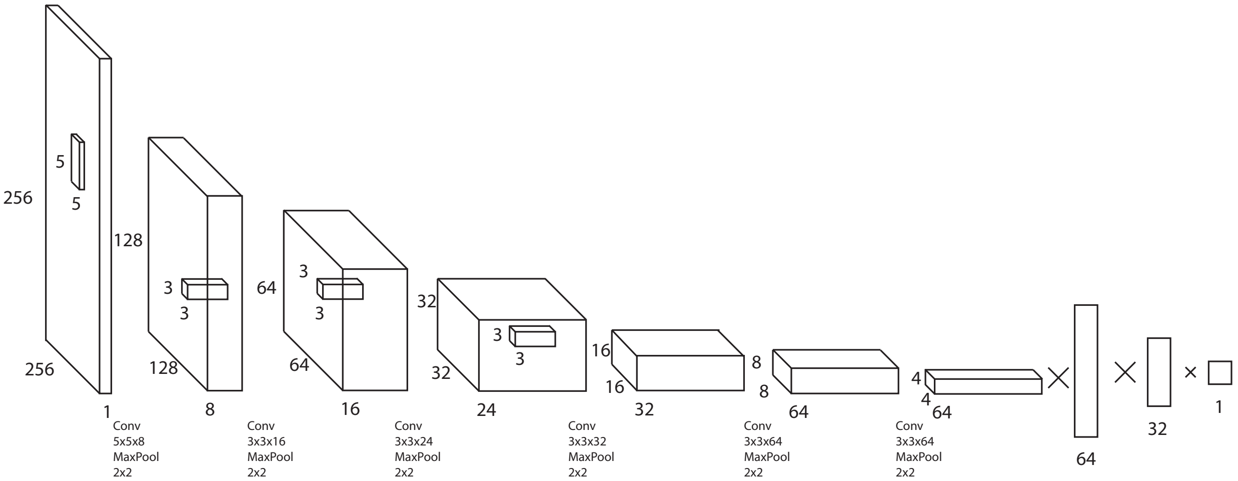


Figure 2. Network architecture

IV. Convolutional neural networks for image analysis

The first appearance of convolutional neural network was Yan LeCun's paper [6]. Unlike the feed-forward networks there are new types of layers:

- Convolution
- Pooling (maximum, average)

Convolution layers are based on mathematic convolution operators, but all weights of convolution are trainable.

Pooling layer are used to reduce feature space to prevent overfitting and speed up network training process.

V. Network architecture

We suppose to divide vertebrae detection algorithm into two steps:

- Train classifier for image patch, that predicts probability the whole vertebra on the patch
- Detection of the vertebrae on the whole image using sliding window method

We choose the following neural network architecture, based on the work [6]:

- Group of convolution and pooling layers following each other
- The last 3 layers are dense and last layer predicts probability that the image contains vertebra

As activation function on all layers we use LeakyReLU [7], on the last layer - sigmoid function [8].

Neural network architecture is shown on figure 2.

VI. Implemenation and training

For network implementation we use Google Tensorflow library [9]. The library allows to develop different neural network types, providing support for a large number of layers (convolution, pooling, dense, activation and more) and architectures (feed-forward, recurrent networks).

Training images have been chosen by specialist (neurosurgeon) from Gomel regional clinical hospital archive, anonymized and labeled.

Since the images amount is small (about 160), for network training we have used different data augmentation approaches:

- Rotation $(90^\circ, 180^\circ, 270^\circ)$
- Reflection (vertical and horizontal)
- Noise addition

For network we use rectangular patches with size 256×256 pixels, because this size guarantees the patch will contain whole vertebra despite of vertebra's rotation angle.

We doesn't scale any patch, because it can change vertebra geometrical parameters and it isn't acceptable for our task.

Image samples used for training are shown on figure 3.

Since spine images contain a few patches with vertebrae, we have chosen all the images with vertebra and some images without vertebra, so they are splitted with $1:1$ ratio.

We split the whole image set into three parts - train, validation and test. 80% of images are in the train part, 10% - in validation and 10% - in test, and we train network for 40 epochs.

Classification accuracy on the training set is growing from 50% to 98%, on validation set from 50% to 94%. We don't increase epochs count because it cause overfitting.

After training accuracy on test set is 93%. We divide all the X-ray images into rectangles and predict probabilities of vertebra position in the rectangle.

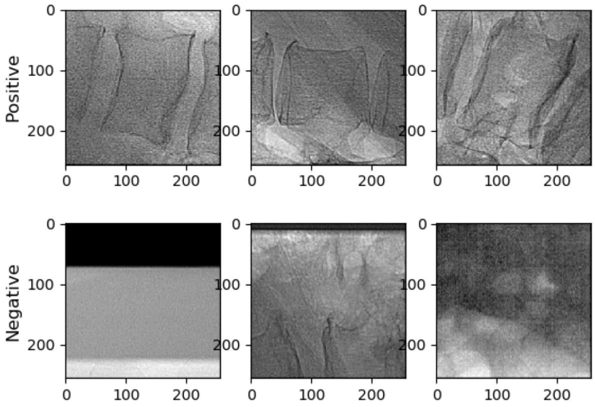


Figure 3. Positive and negative image samples

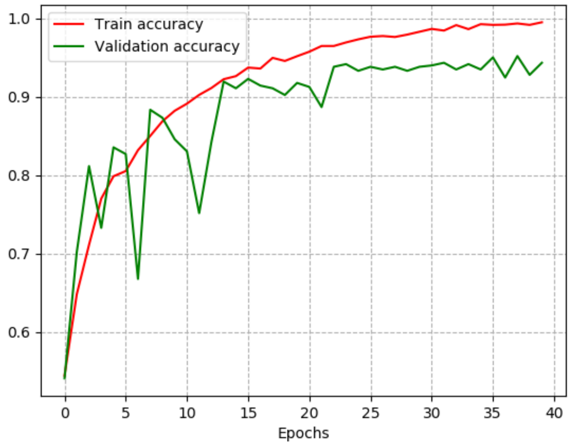


Figure 4. Accuracy chart

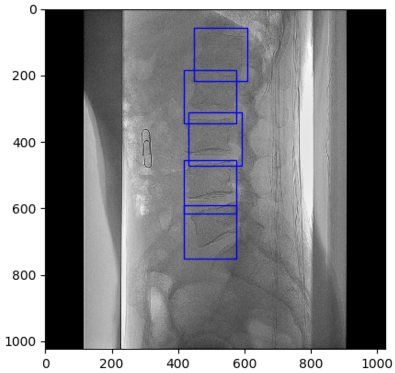


Figure 5. Output of algorithm

VII. Conclusion

We developed and trained neural network for X-ray image vertebrae detection. The network shows good result and can be used for vertebrae detection to calculate their geometric parameters, which can be used for further automated diagnosis of diseases of the musculoskeletal system.

References

[1] K. S. Kurachka and I. M. Tsalka, “Building three-dimensional vertebrae model of lumbar spine based on computer tomography images,” in *Information Technologies and Systems*, 2015, pp. 200–201, (in Russian).

[2] E. L. Tsitko, A. F. Smeyanovich, E. S. Astapovich, and E. V. Tsitko, “Roentgenometric analysis of the kinematics l4-l5 and l5-s1 spine segments at the third stage of the degenerative process,” *Novosti Khirurgii*, vol. 23, no. 2, pp. 202–208, Mar-Apr 2015, (in Russian).

[3] C. H. Huang, “A fast method for spine localization in x-ray images,” in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2013, pp. 5091–5094.

[4] F. Lecron, M. Benjelloun, and S. Mahmoudi, “Fully automatic vertebra detection in x-ray images based on multi-class svm.” in *Medical Imaging: Image Processing*, 2012, p. 83142D.

[5] A. Mehmood, M. U. Akram, and A. Tariq, “Vertebra localization and centroid detection from cervical radiographs,” in *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, March 2017, pp. 287–292.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[7] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[8] J. Han and C. Moraga, *The influence of the sigmoid function parameters on the speed of backpropagation learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 195–201.

[9] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

Temporal data retrieval efficiency

Michal Kvet, Karol Matiaško
Department of Informatics, Faculty of Management Science and Informatics
University of Žilina
Žilina, Slovakia
Michal.Kvet@fri.uniza.sk, Karol.Matiasko@fri.uniza.sk

Abstract— **Current databases are characterized by significant data amount to be managed, processed, stored and retrieved. Temporal extension borders individual states by the validity allowing monitoring evolution over the time with emphasis on analysis, decision making and optimization. Essential part of the processing covers just the data storage management, quality and effectivity of the data retrieval based on physical and logical architecture as well. In this paper, individual temporal architectures are proposed to be handled by temporal extension of the Select statements. It reflects not only new clauses of the command itself, but also change of the optimizer access to the data and temporal index as well.**

Keywords—temporal database; temporal retrieval; indexing; HWM; Select statement extension

I. Introduction

Data characteristics and values change more or less frequently. States reflected by attribute values are changed over the time, even structure or data types can evolve, become more specific and accurate. Information systems should be aware of them and provide sufficient power for proper, fast and consistent reactions. Data are commonly stored in the database, which forms the core of the information system data management. Conventional approach used today is based on paradigm of storing only current valid data forming actual image. Such approach is, however, not suitable for intelligence aspect of information systems. Therefore, temporal paradigm concept has been proposed. Database management is shifted to the temporal approach, thus, data changes and evolutions are monitored over the time to provide complex information. It contributes to a comprehensive opportunities for decision making either in the context of operational decisions, but also long-term aspect as well [7] [12].

Temporal database approach is based on temporal paradigm, which extends the definition on time spectrum. Object itself is therefore not defined only by its identifier, but also time validity component is present. Unfortunately, temporal paradigm has not been approved by standardization organizations yet, approaches are not part of common database management definition (like Oracle, MySQL, PostgreSQL, etc.) resulting in total abolition of the development in that field [1]. As a result, complex solution, individual processes, data modelling and management must be done on your own without complex environment possibilities [5] [9].

This paper deals with current temporal database architecture overview covered by section 2. Physical architecture, representation and performance limitations are discussed in section 3. Data retrieval process and temporal clause extensions are described in section 4. Also performance evaluation is part of that section.

II. Temporal architectures

Conventional database approach is based on storing and managing only current valid states, which are overwritten at the time of change. Although historical data can be found as part of transaction data logs, it is inevitable to mention the effectivity of such approach as well as impossibility for future valid state management [2]. Moreover, transaction log files are not long-term stored automatically, thus another layer for dealing with such data had to be developed [2]. Moreover, during the data retrieval, such data had to be loaded, parsed by composing consistent image. It lasted too much time, therefore, such way was stagnant.

Temporal paradigm concept provides completely new approach. First temporal architectures themselves were proposed in 80ties of the 20th century. They were more or less based on temporal extension of current approaches by adding temporal layer. As the main approach of the time component processing can be mentioned object level validity model [8] [9] [14]. Solution is based on extension of the primary key by using validity elements modelled either by one attribute (each new state delimits the validity of previous state), by time interval (two attributes defining starting and end point of the validity with emphasis on closed or open representation). The last solution is based on defining starting point of the objects´ validity and duration modelled by proposed interval data types.

Several temporal aspects can be managed, like validity, temporal locality, transaction time validity, etc. In that case, individual processed spectrum limits the object level model – uni-temporal (one time spectrum is processed, usually validity), bi-temporal (two time spectrums are managed). Generally, multi-temporal solution can be used. Fig. 1 shows the model of object level temporal solution.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

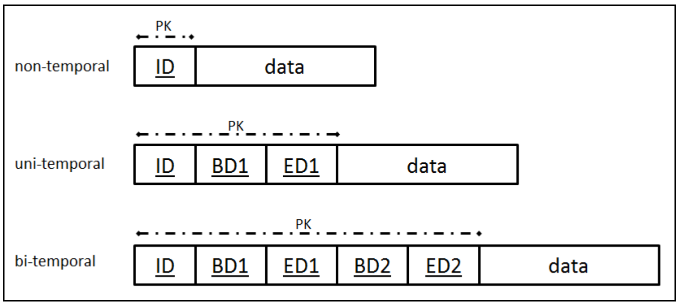


Fig. 1. Temporal models (object level approach) [9]

Notice, that it is based on primary key extension, which significantly influences the possibility for referential integrity definition, whereas complete primary key is copied as foreign key. In this case, it is always composite. Moreover, individual time intervals must be covered. To remove such limitation, several approaches have been proposed, the most essential to be mentioned are these [9] [17]:

- Non-current data are located in separate layer, either modelled by using nested tables (table as part of the current definition) or by using separate tables (in that case, referential integrity is based on current state, which must be, however present, otherwise it is not possible to create referenced object. Moreover, interval validity must be highlighted and covered).
- The second approach uses the fact, that the foreign key may not be limited to the primary key, but generally to a unique index. Thus, another attribute is defined holding unique values, usually provided by sequences, respectively autoincrement.

Our proposed attribute oriented architecture shifts the granularity of the processing to attribute itself. Thus, individual change is not defined as the whole state update, but it is defined for attribute. Thus, individual state is created as composition of particular attributes during the defined interval or timepoint. It uses three layer architecture. Current valid states are located in the first one, thanks to that, applications dealing with only actual states can work without any change. Core of the attribute oriented approach is just the temporal layer storing all changes based on temporal attribute (notice, that not all attributes must be monitored over the time). The last layer consists of data, which are not currently valid – historical and future valid images. Also, this layer is attribute oriented. Fig. 2 shows the architecture. External processes and applications cannot communicate with non-current states directly, whole communication is done by using temporal layer. Yellow arrows express the communication technologies and opportunities. If existing applications are defined, they can directly communicate with actual valid state layer. Rest processes query temporal layer, which can access any object. Communication separation improves reliability and ensures consistency of the system. Principles, data management and operations are described in [13] [15].

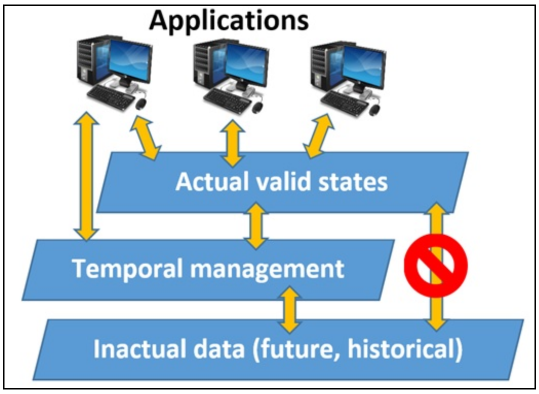


Fig. 2. Temporal models (object level approach) [13]

Extension of the attribute oriented approach in terms of performance is provided by attribute grouping. Previously described solution deals with attribute itself. Many times, some data portions are synchronized, when change occurs. In that case, each attribute change would require inserting new row into temporal system layer. Group management automatically detects synchronization and creates groups, which are managed as one attribute. Thanks to that, there is significant performance benefit [13]. Solution is described in [13] [14].

Special category covers spatio-temporal systems. Individual states are delimited not only by time spectrum, but also by location [3] [4] [6] [10]. Whereas object cannot be located in two places at the same time, proposed attribute oriented approach is robust and can be directly used also for spatio-temporal modelling.

Most current solutions are based on conventional paradigm extension and do not provide specific solution for temporal queries.

III. Physical Database Architecture and Performance Limitations

Database server can be divided into two parts – instance and database covering physical files. The interlayer between instance (memory structures) and database is just tablespace, which separates physical structures from the database management. Thanks to that, user does not need to know physical locations and approaches to the data themselves. Data in the database are located in the data files formed into the blocks, which are grouped into extents. And it can be the bottleneck of the system performance. During the data retrieval, system does not directly know about the availability and fullness of the block. Thus, each one must be loaded into memory (buffer cache) and subsequently evaluated. Moreover, database system does not even know directly, if the block is completely empty. It significantly degrades performance, if no accelerated retrieval of requested data is available. Solution performance is based on using index structure.

Index is an optional structure associated with a table or cluster, which can speed data access (process of data retrieval). By creating an index on one or more columns of a table, you gain the ability to retrieve a small set of randomly distributed rows from the table quickly. During individual destructive *DML* statement execution, index is built, respectively reconstructed. It contains locators to the physical structure on the leaf layer – pointers to the physical files – *ROWID*s. Most often used index structure is just B+ tree index as a default type used in database. Table row identifier (*ROWID*) and associated column values are stored within index blocks in *balanced tree structure*. Important property of such index is the fact, that it cannot manage *NULL* values at all [16].

It is formed by *root node, internal nodes* and *leaf nodes* consisting of *ROWID* pointers. Data on the *leaf layer* are ordered and connected via double directional linked list. Such index can be used also in temporal environment, if some pre-requisites are met.

Data access during the retrieval can use proposed index, if the appropriate and suitable one is defined. It shifts the *Table Access Full* method, where all blocks are loaded into memory and evaluated to *Index Scan*, which uses defined index (which is commonly already loaded in the memory) as pointers to the particular data blocks. As a result, it is not necessary to scan all blocks covered by *HWM* (the last used block associated with particular object – table) [9] [10].

A. Managing undefined values

Undefined value management is one of the most significant problem temporal database must face. Unknown values are commonly marked as *NULL*. Problem of *NULL* value is just impossibility to be managed using majority of indexes. Thus, if the value can hold *NULL*, particular index cannot be used shifting the access method to *Table Access Full*. Even worse problem can occur, if *NULL* value should be used for time definition. Time spectrum is usually part of the primary key, as shown in the section 2. In object level temporal system, it is even impossible from the definition point of view. Attribute oriented approach as well as group management does not have such strict limitation, because they use artificial primary key obtained by the sequence. Nevertheless, it is strictly necessary to take emphasis on that fact. Undefined validity is typical problem of temporality. Object states evolve over the time, their changes occur asynchronously. Thus, actual state cannot be usually bordered from the right site of the time interval (end point of the validity is currently unknown). Although *NULL* value can be used, it is not indexed, thus when dealing with current valid states, *Table Access Full* method would be necessary to be used. Imagine the number of non-actual states. Such solution would be really inappropriate. Historical states are robust, but current and future are not. Moreover, what does *NULL* value denote? Absolutely nothing about the timepoint of the validity. It is not even possible to detect, whether such timepoint has occurred or not [15] [16]. In the recent past, we have proposed special notation called *MaxValueTime* expressing undefined validity. It is clearly known, that such moment has not occurred yet – it will be in the future, but the strict time limitation is now undefined. Physically, it is represented by the pointer to the specific memory location dealing with undefined values. Such location is characterized by the array of specific symbols for undefined value of particular attribute delimited by the data types. Thus, each data type, which can hold undefined value, is registered in that memory location and undefined value itself is not delimited *NULL* value, but pointer to such location is used instead.

Process of the undefined value management is following. If the attribute can hold undefined value, during the table definition, particular attribute must be registered by calling *undef_reg* method of the *dbms_temporality* package. It has three parameters, the last one is optional – if not listed explicitly, current user data are used:

- *dbms_temporality.undef_reg:*
    - name of the attribute to be registered (*atr_name*),
    - name of the table, which contains such attribute (*tab_name*),
    - name of the schema (*schema_name*, default – current user).

Internally, such method gets the data type of the appropriate attribute using the data dictionary. Afterwards, it provides registration. In this step, two situations can occur – particular data type has not been registered in the memory yet. In that case, array is extended and new element is added. Binding using pointers is activated, so the undefined value pointer can be used to particular memory. Another situation occurs, if such data type has been already registered. In that case, binding is created to existing element, no new one is created. Moreover, information about the registration is permanently stored in the temporal database registration module – special temporal physical table. Such specific memory location is created from the temporal database registration module automatically during the instance creating process and is permanently located in the memory, until the instance is closed. Fig. 3 shows the principles of registration and activation. The first part deals with the registration itself – user requests the database instance to create element for undefined value, *dbms_temporality* package is invoked, registration is done. The second part deals with the undefined value modelling itself. As you can see, *NULL* value is automatically replaced. In this step, three approaches can be distinguished:

- *force* – it forces using undefined value pointers instead of *NULL* values. Therefore, it will raise error, if the registration is not done and *NULL* value is to be inserted.
- *automate* – if the particular undefined value pointer is not created, it will register it automatically and consecutively replace *NULL* value with undefined value pointer.
- *ignore* – in this case, if possible, *NULL* value is replaced by the undefined value pointer. Otherwise original value will be used.

Such approach can be used for any data type, which values should be monitored over the time.

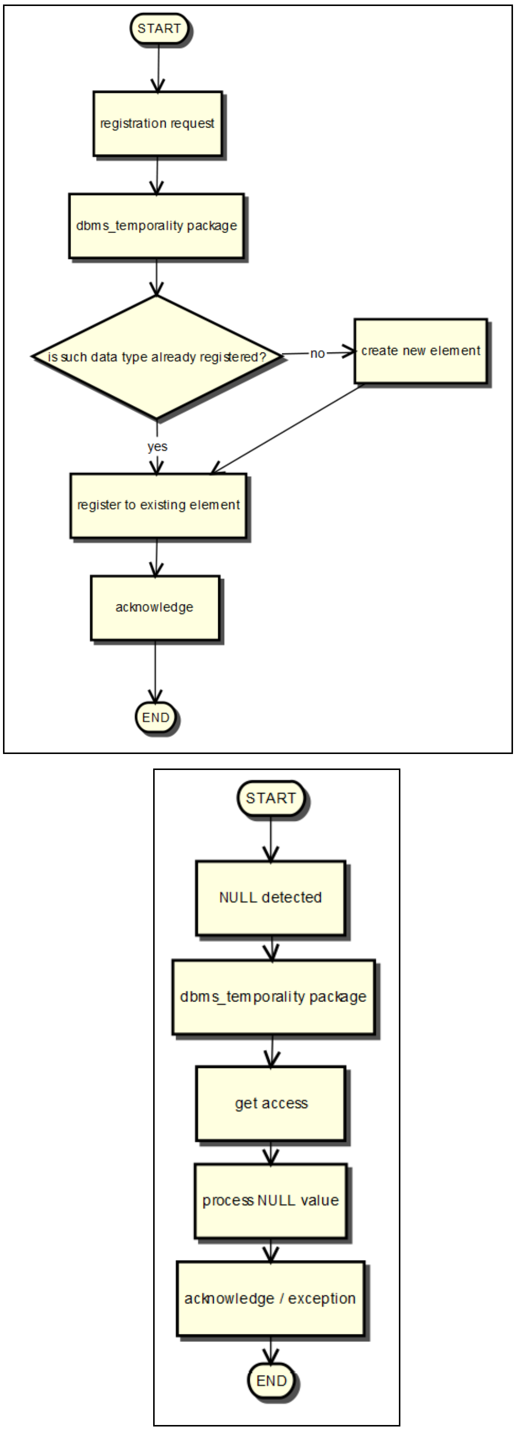


Fig. 3. Registration

IV. Data Retrieval

Data retrieval process is the transformation of the defined query to the result set by *parsing*, *binding*, *executing* the command itself and by providing result set in the *fetch* phase [2] [16]. In this paper, we extend such defined sequence by adding the first *preprocessing step*, which can transform defined query to optimize performance and speed up response of the system. We also add new clauses for the *Select* statement. In this step, it is transformed to existing syntax. Thanks to that, condition evaluation and transformation is done automatically with emphasis on undefined value management as well as time spectrum. Fig. 4 shows the sequence of the processes.

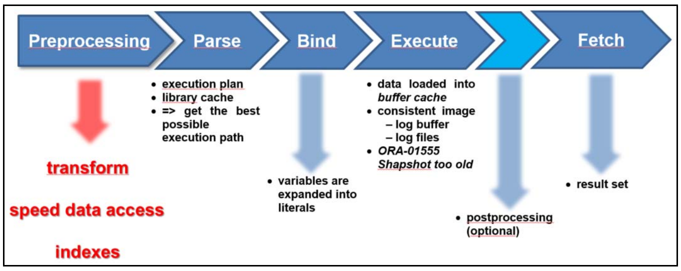

Preprocessing
Parse
Bind
Execute
Fetch
transform
speed data access
indexes
• execution plan
• library cache
• => get the best possible execution path
• variables are expanded into literals
• data loaded into *buffer cache*
• consistent image
– log buffer
– log files
• *ORA-01555 Shapshot too old*
• postprocessing (optional)
• result set

Fig. 4. Query execution steps

Preprocessing layer is another element of the data processing evaluation and retrieval. Whereas database itself does not manage *NULL* values explicitly due to performance limitations, such values are replaced by the pointers to memory location for dealing with undefined attribute values. Afterwards, new clauses are recognized and transformed to the existing syntax.

We propose following clauses as extension of the *Select* statement in temporal environment:

A. *Granularity_monitoring*

When dealing with temporal data, it is necessary to choose the way result set is composed. It is possible to monitor changes as the whole state of the objects (*object*) or to monitor only changes of the attribute themselves (*attribute*). If *object* selection is used, composition of the complex state is done from the temporal layer. If there are also conventional attributes (which changes are not monitored over the time), such attribute values will be marked as *unknown*. Vice versa, *attribute* approach forces the system to evaluate and list only attributes, which values have been changed during the particular *Update*. The rest ones are left unchanged. Thus, conventional attributes are not listed at all. Special category covers the *database* granularity. In that case, the object is not processed as image of the table row, but also dependent objects are taken into account. Complex state is therefore comprised from the multiple table images. Depth of the spreading can be set in optional parameter *depth*. If not listed explicitly, all dependencies are processed.

```
Granularity_monitoring({object | attribute
                       | database[(depth)]})
```

B. *Time_element*

Time element clause defines time interval to be evaluated. In principle, interval is defined by two attributes limiting first and last point of the validity with emphasis on interval representation (interval representation limits, whether first or last point are part of the interval to be evaluated or not [9]). Another category covers only timepoint, which can be modelled either as separate clause or by defining interval with only one timepoint duration.

```
Time_element(timepoint)

Time_element(t1, t2, [{CC | CO | OC | OO}])
```

In this pure clause, all changes are monitored and listed in the result set.

C. *Time_periodicity*

By applying this clause, periodicity and time precision of the evaluation can be influenced. Generally, output set is produced for each change. However, it is also possible to create image at precisely defined time points. The first image is created at the starting point of the validity interval and then, always after defined period has elapsed (delimited by *value* parameter, which is extended by the *format* parameter expressing unit of particular value, e.g. second, minutes, etc.).

Naturally, it is not applicable for timepoint setting in *Time_element* clause.

```
Time_periodicity({always
                  | period(value, format)})
```

D. *Monitored_attributes*

Temporal layer record attributes are evaluated and processed over the time, generats triggers, which ensure storing historical and future valid data as well. These attributes are monitored and whole spectrum of their values can be obtained. However, many times, it is not necessary to react to any change of temporal attribute. In principle, *Select* clause of the statement contains list of attributes, which values must be retrieved in the result set.

```
Select atr1 [, atr2 [, …]] [func1() [, func2
[,…]]]
```

Select clause attribute list can contain also conventional attributes, as well as function calls. It does not need to be the same as *monitored_attributes* list. *Monitored_attributes* list consists of attributes, which change invokes new row in the result set. It can hold only temporal attributes, not functions. Following code shows the logical call of *Monitored_attributes* clause.

```
Monitored_attributes atr1 [, atr2 [, …]]
```

For the evaluation reasons and accessibility, list of attributes is transformed internally into nested table consisting of particular attribute list.

E. *Epsilon*

Data stored in the database can originate in different devices, sensors with various precision, reliability and quality [11]. These data are produced in second fractions and even more often, therefore to compact result set, another clause is introduced. *Epsilon* clause allows you to define minimal difference of consecutive values, which are relevant and should be placed into result set. Parameter of such clause is a set of pairs (*value – minimal difference*). *Value* itself can be produced directly as attribute value, but can be also output of the function covering multiple (temporal) attributes together. Thanks to that, robust solution is proposed.

```
Epsilon(val1, min_dif1 [val2, min_dif2 [,…]])
```

For the simplicity of description, *val* represents attribute or function call. Such clause can be also used during data loading to remove not relevant changes.

As already mentioned, proposed characteristics provide another layer providing developer easier query coding. Evaluation is done automatically and database optimizer gets *Select* statement transformed into existing syntax. When dealing with the performance, we can come to the conclusion, that there is no significant impact, when preprocessing module is added. Regarding the system sources, no changes were recorded mainly because of the low costs of the solution as well as the amount of data to be processed in temporal environment. Time processing requirements are also increased in minimal manner – less than 0,5% for 1000 sensors. In principle, with an increase in the data amount, the impact of added preprocessing layer decreases.

Fig. 5 shows the dependence between data amount (number of sensors providing data) and processing in added layer. As you can see, the difference is really minimal in comparison with benefits, our proposed solution offers. *X* axis models number of sensors to be evaluated, *Y* axis represents performance degradation, expressed in percentage.

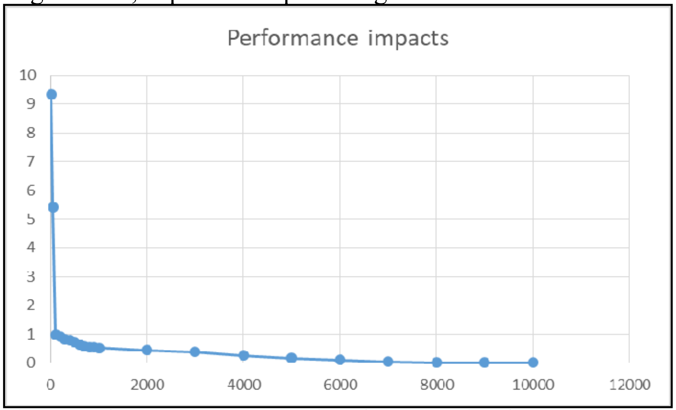


Fig. 5. Performance impacts – number of sensors and performance

For 10 sensors, performance degradation is 9,32%, for 500 sensors, it reflects 0,73%. If the number of sensors is higher than 3000, performance impact is lower than 0,1%.

These results were provided using Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 – 64 bit Production; PL/SQL Release 11.2.0.1.0 – Production. Parameters of user computer are: Intel Xeon E5620; 2,4GHz (8 cores), operation memory: 16GB and HDD: 500GB. Number of sensors, which data should be managed evolved.

V. Conclusions

Core of the current information systems is just data processing and management. Its intelligence is markedly supported by decision making. However, this requires reliable, correct and comprehensive data. Temporal database allows you to manage data over the time in the main data structure. In this paper, several architectures have been described with emphasis on granularity. On the one hand, it is object oriented approach, on the other hand, attribute granularity is defined. Interlayer between them is just group management approach, which takes synchronized data changes by forming temporal attribute groups.

Temporal data are characterized by huge data amount and temporal aspect. Our temporal approach is responsible to the transactions and can cover all relational integrity rules with emphasis on performance and efficiency.

Data retrieval is a complex process and covers significant performance aspect. Indexes provide powerful tool for locating data in the physical structure. As described, commonly used indexes cannot manage *NULL* values in the tree, because of the inability to compare them. Therefore, memory module has been proposed, which consists of element array, to which undefined values point. As a consequence, undefined value does not use *NULL* mark, but a special denotation is proposed, which makes it possible to be indexed.

In this paper, also new clauses (*Granularity_monitoring, Time_element, Monitored_attributes, Epsilon*) have been proposed by forming extension of existing *Select* statement syntax placed in temporal environment. Transformation is done automatically before the evaluation by database optimizer itself.

We will focus on spatio-temporal extension in the future and accessible transformations of the *Select* statement by defining architecture itself as well as management of the data processing and evaluations. Our research interest will also be in distributed and parallel environment.

Acknowledgment

This publication is the result of the project implementation:

Centre of excellence for systems and services of intelligent transport, ITMS 26220120028 supported by the Research & Development Operational Programme funded by the ERDF and *Centre of excellence for systems and services of intelligent transport II.,* ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

This paper is also supported by the following project: "*Creating a new diagnostic algorithm for selected cancers*," ITMS project code: 26220220022 co-financed by the EU and the European Regional Development Fund.

"Podporujeme výskumné aktivity na Slovensku
Projekt je spolufinancovaný zo zdrojov EÚ

References

[1] K. Ahsan, P. Vijay. “Temporal Databases: Information Systems”, Booktango, 2014.

[2] L. Ashdown. T. Kyte “Oracle database concepts”, Oracle Press, 2015.

[3] G. Avilés et all. “Spatio-temporal modeling of financial maps from a joint multidimensional scaling-geostatistical perspective”, 2016. In Expert Systems with Applications. Vol. 60, pp. 280-293.

[4] R. Behling et all., “Derivation of long-term spatiotemporal lanslide activity – a multisensor time species approach”, 2016. In Remote Sensing of Environment, Vol. 136, pp. 88-104.

[5] C. J. Date, N. Lorentzos, H. Darwen. “Time and Relational Theory : Temporal Databases in the Relational Model and SQL”, Morgan Kaufmann, 2015.

[6] M. Erlandsson et all., “Spatial and temporal variations of base cation release from chemical weathering a hisscope scale”. 2016. In Chemical Geology, Vol. 441, pp. 1-13

[7] J. Janáček and M. Kvet, “Public service system design by radial formulation with dividing points”. In Procedia computer science [elektronický zdroj], ISSN 1877-0509, Vol. 51 (2015), pp. 2277-2286

[8] T. Johnston. “Bi-temporal data – Theory and Practice”, Morgan Kaufmann, 2014.

[9] T. Johnston and R. Weis, “Managing Time in Relational Databases”, Morgan Kaufmann, 2010.

[10] A. Kadir and N. Adnan, “Temporal geospatial analysis of secondary school students´ examination performance”, 2016. In IOP Conference Series: Earth and Environmental Science, Vol 37, No. 1.

[11] M. Kvassay, E. Zaitseva, J. Kostolny, and V. Levashenko, “Importance analysis of multi-state systems based on integrated direct partial logic derivatives”, In 2015 International Conference on Information and Digital Technologies, 2015, pp. 183–195.

[12] M. Kvet and J. Janáček, “Relevant network distances for approximate approach to the p-median problem. In Operations Research Proceedings 2012: Selected Papers of the International Conference of the German operations research society (GOR)”, Leibniz Univesität Hannover, Germany, Springer 2014, ISSN 0721-5924, ISBN 978-3-319-00794-6, pp. 123-128.

[13] M. Kvet, K. Matiaško, “Transaction Management in Temporal System”, 2014. IEEE conference CISTI 2014, 18.6. – 21.6.2014, pp. 868-873

[14] M. Kvet and K. Matiaško, “Uni-temporal modelling extension at the object vs. attribute level”, IEEE conference UKSim, 20.11 – 22. 11.2014, , pp. 6-11, 2013.

[15] M. Kvet and K. Matiaško, „Temporal transaction integrity constraints Management“, Cluster Computig, Volume 20, Issue 1, pp. 673–688

[16] D. Kuhn, S. Alapati, B. Padfield, “Expert Oracle Indexing Access Paths”, Apress, 2016.

[17] A. Tuzhilin. “Using Temporal Logic and Datalog to Query Databases Evolving in Time”, Forgotten Books, 2016.

Logistic conception for real-time based info-communication system applied in selective waste gathering

István Lakatos,
Department of Automotive and Railway Engineering
Szechenyi Istvan University
9023 Gyor, 1 Egyetem square, Hungary
lakatos@sze.hu

Tamás Péter,
Department of Control for Transportation and vehicle system
Budapest University of Technology and Economics
1111 Budapest, 2 Stoczek street, Hungary
peter.tamas@mail.bme.hu

Ádám Titrik,
Department of Automotive and Railway Engineering
Szechenyi Istvan University
9023 Gyor, 1 Egyetem square, Hungary
titrika@sze.hu

Decreasing of available raw materials, fossil energy source and new approach made recycling necessary and inevitable. For recycling selective waste areas have been placed, where inhabitants can drop their waste for free. Waste containers have different saturation time, while gathering process is predefined, therefore independently whether it is needed or not they will be emptied. Further problem is that there is no optimization on the way to the container, which results in serious environmental load (noise and air pollution). This study deals with high level route optimization to minimize environmental load concerning noise and air pollution and by the application of real-time communication the appropriate containers are only going to be emptied.

***Keywords:** route optimization, real-time system, info-communication, waste collection*

I. INTRODUCTION

In order to decrease the environmental impact and to emphasize the importance of recycling, selective waste collection islands have been created in different parts of the towns where the inhabitants can place their selective waste free of charge. It is a serious logistic task, more exactly an inverse logistic task to collect this waste. It is important to find the optimized route as well as to detect the emptying of which waste container is reasonable. The present study is describing an info-communicative and an inverse logistic process development as possible solutions for this double task, with the help of which the efficiency of waste collection can be maximized.

II. DESCRIPTION OF SOLUTIONS APPLIED IN OUR DAYS

In their study, Byung–In Kim, Seongbae Kim, Surya Sahoo [1] apply time-windows in waste collection in favour of development, moreover, they present graphs and algorithms bearing optimization in mind, with which they report savings of nearly 10 % on the whole collection route. It means that the collection process could be reduced from 10 9-hour long vehicle rounds to 9 9-hour long vehicle rounds. In further articles, on the basis of similar examinations the length of the collection route was decreased by 24.6 %, besides waste collection takes 44.3% less time, which was carried out in Turkey [2]. Studies exist that fuel utilization, gathering route and time were decreased by software application fit to local conditions – GIS 3D modelling, ArcGis®, and RouteViewPro™ [3].

In Hungary the Bay-Logi Research Institute commissioned by AVE Miskolc Ltd. is redesigning the waste transportation processes with the help of RouteSmart, a special GIS-logistic software in order to decrease the distances taken by waste-collecting vehicles and thus to cut down operational costs. As a further solution the experimental town has provided an opportunity to report the emptying of waste in case of filling-up.

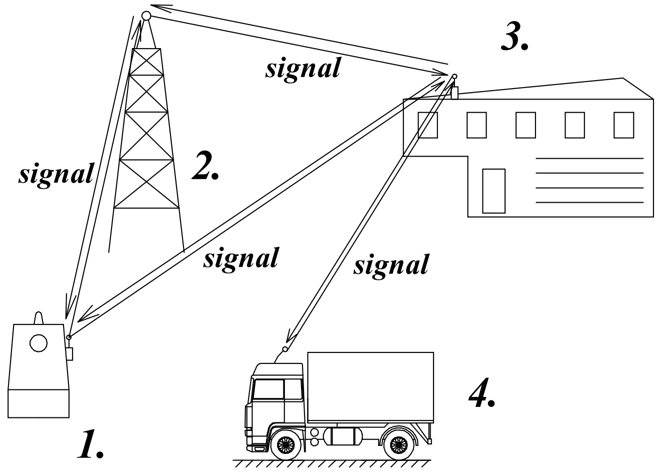
On the one hand, the presented systems have all contributed to the definition of optimized route, but on the other hand, these solutions could not find out the emptying of which containers is reasonable. The aim of this study is to improve the inverse logistic processes, using info-communication, in order to enhance the efficiency of collection.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

III. Description of the real-time based info-communication system

The basis of the operation of the system is real-time based info-communication and GIS technology [4, 5, 6, 7]. Real-time based info-communication takes place among container – centre – vehicle, thus exact information of which containers should be emptied can be gained at any point in time.

The elements of the real-time based info-communication selective waste-collection are the following (figure 1):



1. selective waste- container,
2. signal tower,
3. route-planning centre,
4. waste-collection vehicle.

Fig. 1. Elements of real-time based info-communication system and signal flow

The sign provided by the container, which can be at reaching the given level, or at a time retrieved arbitrarily, is taken by the centre planning the drives. The sign can then be forwarded to the vehicle on collection route, thus the given route can be redesigned during the collection. Exact emptying plans can be compiled with the help of this development.

IV. Designing process of the optimisation of real-time based info-communication waste collection

In order to enhance the efficiency of waste collection the savings method has come to the fore in the advantages of different kinds of routes defined in logistics and during the design of rounds. Real-time based waste collection makes continuous redesign possible while keeping in mind appropriate viewpoints. There is a two-level route definition, that is continuous optimisation is carried out before and during the collection, and the following main considerations are taken into account (Figure 2):

- taking-up reasonable containers
- optimisation of fullness of vehicles
- route optimization

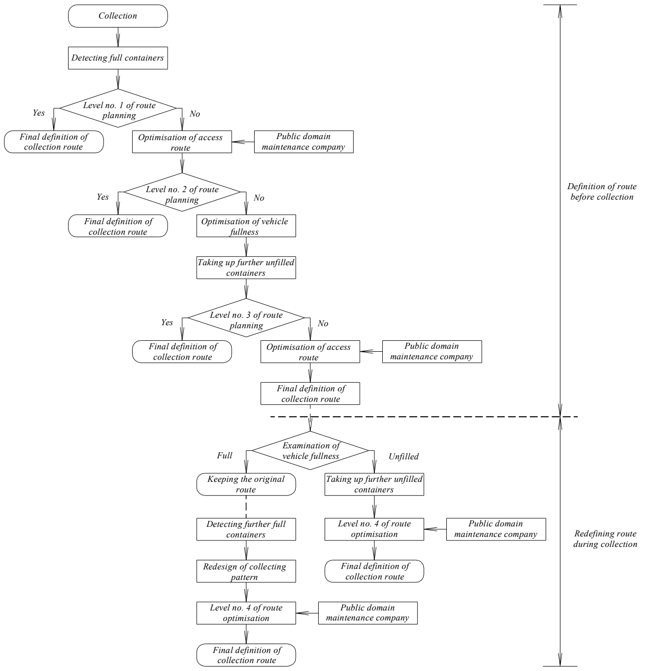

Collection
Detecting full containers
Level no. 1 of route planning
Yes
No
Final definition of collection route
Optimisation of access route
Public domain maintenance company
Level no. 2 of route planning
Yes
No
Final definition of collection route
Optimisation of vehicle fullness
Taking up further unfilled containers
Level no. 3 of route planning
Yes
No
Final definition of collection route
Optimisation of access route
Public domain maintenance company
Final definition of collection route
Definition of route before collection
Examination of vehicle fullness
Full
Unfilled
Keeping the original route
Taking up further unfilled containers
Detecting further full containers
Level no. 4 of route optimisation
Public domain maintenance company
Redesign of collecting pattern
Final definition of collection route
Level no. 4 of route optimisation
Public domain maintenance company
Final definition of collection route
Redefining route during collection

Fig. 2. Flow chart of high level waste collection

Efficiency in waste collection can be maximized by blending optimisation in certain areas during which further advantages can be gained in environment protection and traffic safety. During optimisation external INPUTS have been available for help in the process – cooperation with the public domain maintenance company – and the optimisation of collection does not only concern the given collection round (with the help of statistical data the containers not yet full but expected to be filled soon are taken in the emptying process), but it also optimises the whole of the collection process in more days.

Testing of real-time based info-communication waste collection among real circumstances

In connection with the introduction of the system, examinations have been carried out on the basis of the data provided by the experimental town. In the examinations runs were completed by using the registered data. The stimulation of the new collection method has been carried out with the help of the PARAGON software, on the basis of the following data:

A. Waste-collection examination within the experimental town

Data characteristic of the interval examined:

- type of waste collected: PET bottles,
- number of containers collected in the period of 5 days: 755 pieces
- covered route length: ~878 km,
- capacity of containers applied: 1,1 m^3,

in a waste-collection island a piece of container has been used for one type of waste.

Visiting all the containers:

In the 5-day long interval chosen arbitrarily 755 containers were emptied during the waste collection which could have been achieved by covering 878 km (Table 1). Before planning the route the fullness of containers is not examined, so all the containers – independently of whether their emptying has been reasonable or not – are looked for. On the basis of the data 135 of the 755 containers were empty or 0-12.5% full. On the basis of the fullness examination of the presently used 1.1 m^3 container to be opened (filled and emptied) from the top, the empty containers were – supposedly - not lifted up-emptied by the field survey of the reasonableness of emptying.

TABLE I. PLAN OF A FIVE-DAY LONG WASTE-COLLECTION

Day	Covered route length [km]	Number of vehicles used [piece]	Rounds [piece]	Collected containers [piece]
Monday	185	4	8	155
Tuesday	165	4	4	152
Wednesday	171	4	5	149
Thursday	171	4	5	146
Friday	186	3	7	153
	∑878km	∑19pieces	∑24pieces	∑755pieces

The route plans gained during the traditional collection can be seen in the following figures, which have been broken down to days (Figures 3-7).

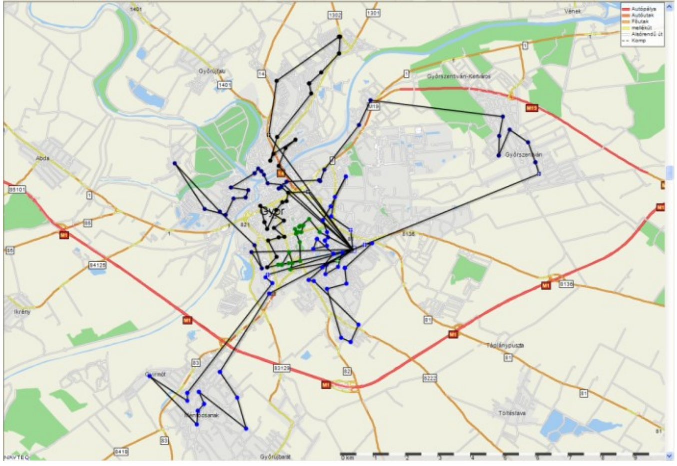


Fig. 3. Traditional collection plan for 155 containers on the 1st day

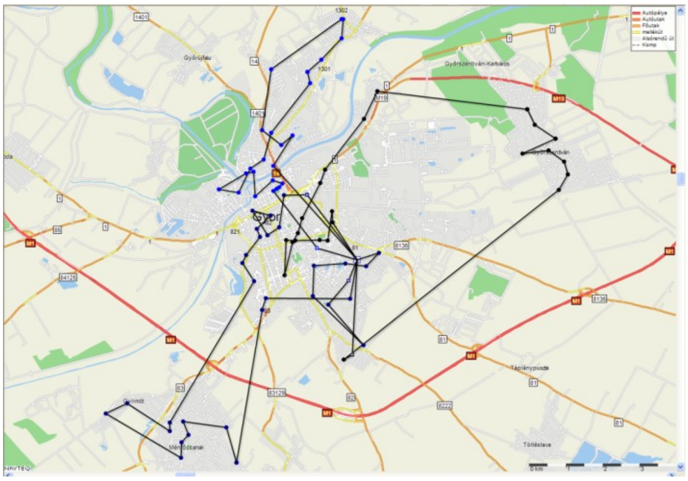


Fig. 4. Traditional collection plan for 152 containers on the 2nd day

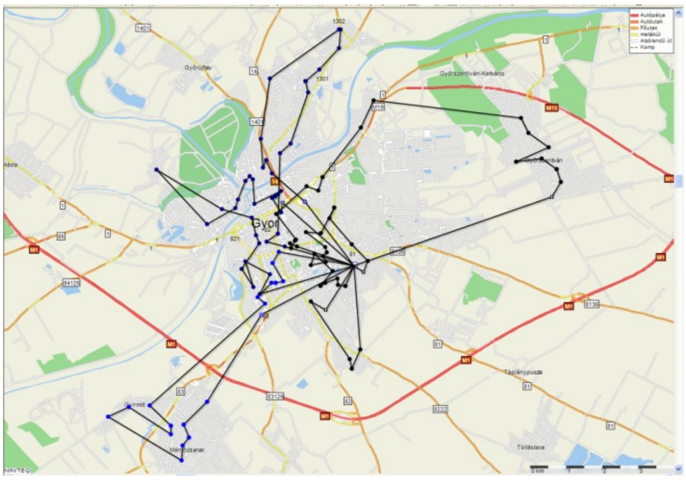


Fig. 5. Traditional collection plan for 149 containers on the 3rd day

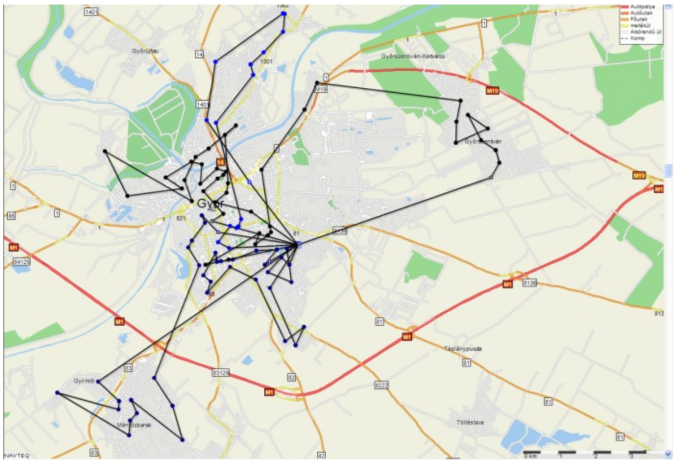


Fig. 6. Traditional collection plan for 146 containers on the 4th day

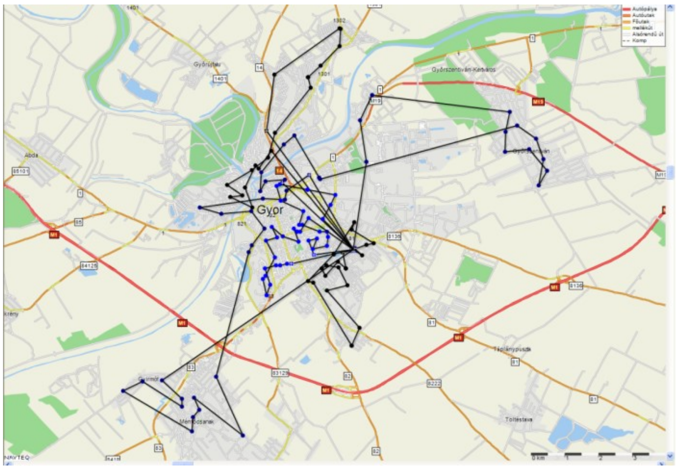


Fig. 7. Traditional collection plan for 153 containers on the 5th day

B. *Emptying carried out with the application of the new system (optimised):*

On the basis of the data, 135 of the 755 containers were empty or 0-12.5 % full. These containers have not been taken up. Moreover, containers which became 100 % full in two consecutive collection events, were drawn together, thus in the first case they were not emptied. On the basis of the optimisation the vehicles covered 795 km and emptied 615 containers (Table 2).

TABLE II. PLAN OF A FIVE-DAY LONG WASTE-COLLECTION APPLYING OPTIMISATION

Day	Covered route length [km]	Number of vehicles used [piece]	Rounds [piece]	Collected containers [piece]
Monday	175	4	8	155
Tuesday	138	3	4	89
Wednesday	150	3	6	121
Thursday	159	3	6	121
Friday	173	3	7	129
	∑795km	∑16 pieces	∑31 pieces	∑615 pieces

The route plans acquired during the improved collection can be seen in the following figures, which have been broken down to days (Figures 10-14).

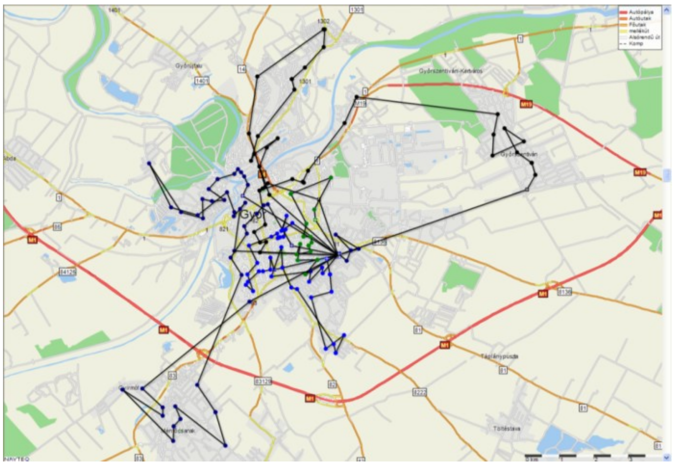


Fig. 8. Optimized collection plan for 155 containers on the 1st day

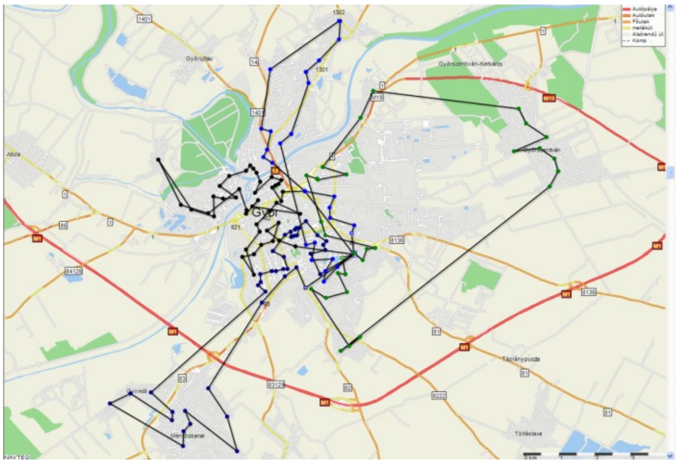


Fig. 9. Optimized collection plan for 89 containers on the 2nd day

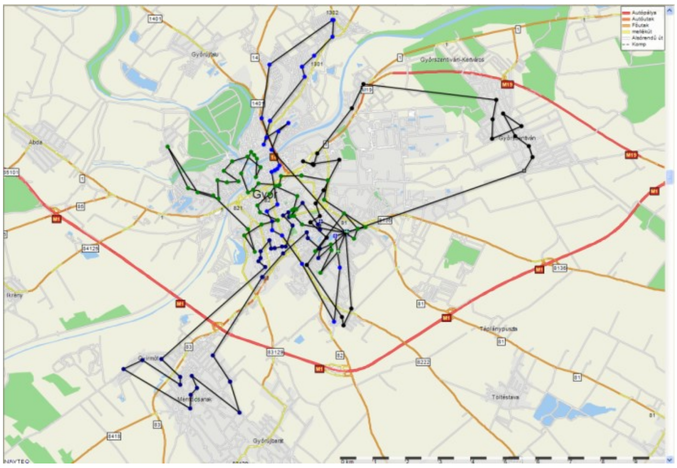


Fig. 10. Optimized collection plan for 121 containers on the 3rd day

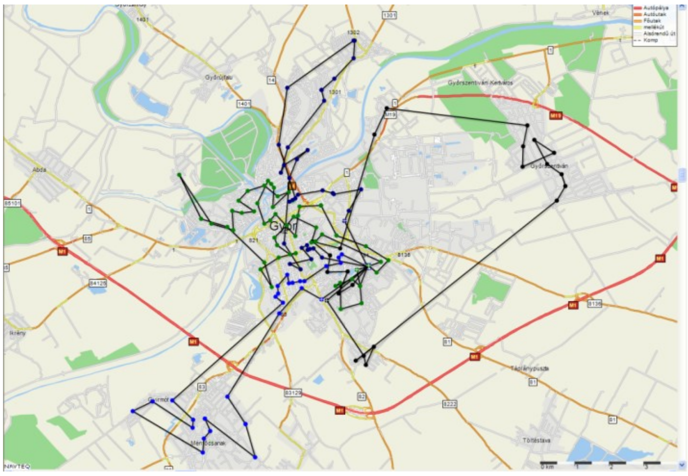


Fig. 11. Optimized collection plan for 121 containers on the 4th day

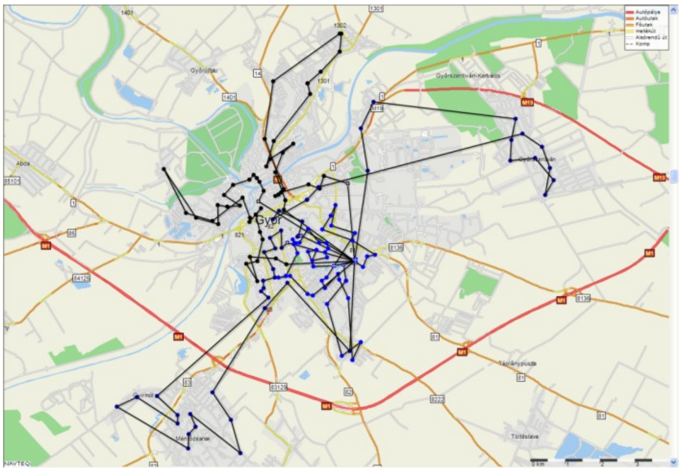


Fig. 12. Optimized collection plan for 129 containers on the 5th day

By examining the data of the charts (tables 1 and 2), in the period of five days an additional route of about 80 km can be saved in the selective waste-collection area of the experimental town, furthermore, 140 pieces of causeless emptying can be avoided.

The following advantages can be drawn up with the basic development of the system – examination of container fullness:

directly:

- decrease of the length of the route by about 10%,
- container lifting-emptying is less by about 20%,
- decrease of maintenance and repair in respect of both containers and vehicles,

indirectly:

- health cost reduction (on the basis of EU examinations, diesel exhaust gas is carcinogenic),
- in traffic safety (less route covered, less risk of accidents).

V. Further possibilities on the field of dynamic optimal control

The research, which is introduced in this paper, contains modern technologies, modelling methods, measurement and software development methods that are appropriate for the integration of the real-time based info-communication system to the selective waste-collection system.

In a more general sense, it can be stated that the effectiveness of the method can be further increased in several fields, if the dynamic stochastic road traffic flows on large urban networks is considered in the research.

It is important to take into account the related real-time control, which is an extremely modern and precise research area. The dynamic network traffic model and its analysis also play an important role in these researches. Dynamic network processes are linked to the tests in such a way that the presented method, along with the space of the applied distances, is also suitable for route optimization in the time of the arrival time parameter if it takes into account not only the data from the traffic manager but also the traffic data from the Central Traffic Controller. This entire process can shorten the time, resulting in minimal traffic interference, which can also be an objective in the discharge of urban communal tasks.

5.1. The possibilites of the agent-based modeling

The studies can be extended to the possibilities of agent-based modeling, in our case, to connect multiple agent systems, and taking into account the co-operation possibilities of transport vehicles. The research is pointing to a multi-agent traffic simulation system. Since the intelligence of individual agents also affects a learning mechanism as a result of communication with the environment, so using artificial intelligence is recommended for further research [8]. In this case, a dynamic system of cooperating vehicles and road crossings cooperating with them is implemented. Modeling involves the use of discrete events and methods that are suitable for describing the transport tasks and the operation of traffic networks. There is also the possibility of using automats and Petri nets at transport nodes and the modeling of road networks that are connecting the nodes.

5.2. Possibilites of game theory, artificial intelligence-based modeling

It is also recommended to approach the problem on game theory, on artificial intelligence based methods. In the case of optimum control algorithms in game theory, the task can be seen as a game theory problem where each transport vehicle and intersection fill in the role of the players. Each player tries to minimize their own cost with the appropriate green marking design, which is best suited to disencumber the road segments and the trajectories that they join. Players' decisions also display a global cost that is the distance from the entire road network optimum. The model can be realized, for example, in a Matlab environment. In this case, implemented route planning algorithms can be integrated into game theory-based traffic management methods. The design of the nonlinear predictive control and optimum reference motion and the examination of the conditions that are necessary to integrate game theory-based traffic management methods into the real environment are need to be determined.

5.3. Direct nonlinear dynamic modeling with the positive system class is taken into account

The first definition of positive systems was given by Luenberger: A positive system is a system in which state variables are not negative [9]. The most of the examined road and transport traffic processes meet this requirement based on the original physical condition of the states. In classical literature, when describing road processes, in most cases general linear system equations are set up and do not exploit the positive properties of the process.

We may think that the properties familiar to general linear systems are true to positive systems without any problems, but this is not the case. The conditions for the controllability and observability of positive systems can not be clearly deduced from the methods known in general systems. The problem is especially true if we require not only a set of conditions, but even the intervening signal. Therefore, the description of road processes as a purely positive system is not a trivial task from a control point of view. The control task in this case means that it is necessary to steer the system from one state to another state so that it is valid during the state transition so that states can not add negative values [10].

For the mathematical modeling of road transport network processes, a special hypermetric structure is provided that describes the internal-internal, external-internal, internal-external and external-external relations between network elements in a network located in a range that is not necessarily a singly connected network [11, 12].

The positive nonlinear differential equation describes the internal and external traffic processes:

$$\begin{bmatrix} \dot{x} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} \langle L \rangle^{-1} & \\ & \langle P \rangle^{-1} \end{bmatrix} \begin{bmatrix} K_{11}(x,s) & K_{12}(x,s) \\ K_{21}(x,s) & K_{22}(x,s) \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} \qquad (1)$$

Where: $\langle L \rangle$ diagonal matrices containing the length of internal sectors:, $\langle P \rangle$ diagonal matrices containing the length of external sectors:

$\langle L \rangle = \langle l_1, l_2, \ldots, l_n \rangle$, $\langle P \rangle = \langle p_1, p_2, \ldots, p_m \rangle$

K_{11} internal-internal connection matrix, K_{12} external-internal connection matrix, K_{21} internal-external connection matrix and K_{22} external-external connection matrix. Physical meaning of matrix elements is connection (transfer) speed.

x Vehicle Density Status Characteristics of Internal Sectors,

s Vehicle Density Status Characteristics of External Sectors,

$$x = \begin{bmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ . \\ x_n(t) \end{bmatrix}, s = \begin{bmatrix} s_1(t) \\ s_2(t) \\ . \\ . \\ . \\ s_m(t) \end{bmatrix}, \dot{x} = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ . \\ . \\ . \\ \dot{x}_n(t) \end{bmatrix}, \dot{s} = \begin{bmatrix} \dot{s}_1(t) \\ \dot{s}_2(t) \\ . \\ . \\ . \\ \dot{s}_m(t) \end{bmatrix} \qquad (2)$$

The generalization of the model has also provided a nonlinear positive differential equation describing the operation of global network processes [13].

In our model $0 \le x_i(t) \le 1$ normalized vehicle density state characteristic is used (i=1,…,n). The combined length of vehicles in one section or sector is divided by the length of the section. This calculation can be applied to parking lots, so parking lots are also generalized in the model

The subject of modeling is a nonlinear positive system.

The "material" flows through the network at varying speeds and with the time-dependent distribution factors marked with $\alpha_{ij}(t)$ are defined. The material is embossed by road vehicles.

Speed depends on vehicle density, maximum limit is limited. In addition, the speed function is influenced by weather, visibility, road geometry, quality and width.

$\beta_{ij}(t)$ hindering the transfer of certain sections $(0 \le \beta_{ij}(t) < 1)$, assist $(1 < \beta_{ij}(t))$.

$0 \le u_{ij}(t) \le 1$ switching function takes into account the effect of traffic lights at the transmission of each phase.

Parallel sections (bands), as well as sections and car parks, give each other a vehicle on the network. This transfer is taken into consideration by a proportionality function $0 \le \gamma ij(t)$ or $0 \le \gamma_{ij}(x_i(t), x_j(t), t)$.

Internal prohibiting automated to operate on the network: No transfer is allowed from j to i if i is full: $x_i(t) = 1 \rightarrow S(x_i(t)) = 0$. No transfer is allowed from j to i if j is empty $x_j(t) = 0 \rightarrow E(x_j(t)) = 0$. By using standardized state properties, these conditions can be easily followed. They provide in the model that we do not take "material" from where there is no (density does not enter negative range) and we do not give it where the density has already reached 1.

The network enclosure is a closed curve „G", not necessarily in the range assayed singly connected. On the external sections that have a direct transfer / take-over relationship with a network section we measure the normalized $0 \le s_i(t) \le 1$ traffic density (i=1,…,m).

The transport model: so-called. "Macroscopic" model. The mathematical model is a nonlinear, non-autonomous positive differential equation system.

Based on the above, the speeds can be extracted directly from the model on any trajectories of the network. The path-time function X (t) can be calculated by knowing the two-variable speed function V (t, X) by solving the following integral equation:

$$x(t) = \int_{t_0}^{t} V(\tau, x(\tau))d\tau \qquad (3)$$

The task requires the solution of a first order nonlinear differential equation with the initial condition X (t0) = x0.

From t1 targeting time X (t) no longer increases, ie, the targeting time T = t1-t0. The length of path X along the trajectory and the time t reaches a path function X (t) to which a time T arrives at the point "B" and this mapping provides J real-function: J: x (t) → T

The model of large traffic networks can therefore be used for real-time route promotion taking into account traffic.

For example, when calculating optimum trajectory, another important guideline is that energy consumption and harmful emissions can be estimated based on expected traffic.

The loading and unloading tasks can be described by a probability variable following an exponential distribution and taken into account in the overall optimization task. Once the unloading has been completed, the operation can be considered complete [14].

VI. Summary

During the application of real-time based info-communication device in waste-collection there is no need for causeless emptying, moreover, reasonable emptying takes place in time, thus by applying the system it becomes possible to create a cleaner, liveable town. With the help of the system, environmental load can be reduced considerably, moreover, traffic safety is increased to a higher level. Statistical data obtained during waste-collection make further developments possible, and confirm the importance of placing further containers. This research-development-innovation defines a new direction in favour of maximizing the efficiency of waste-collection. The use of complex, positive dynamic systems has provided a direct opportunity for modeling and optimizing the process. The new network approach and applied graph helped to better analyze the relationships of network processes and effectively apply them. It can be emphasized, therefore, that the modeling of transport networks and supply chains requires the application of very similar methods [15].

Acknowledgement

The research presented in this paper was carried out as part of the EFOP-3.6.2-16-2017-00016 project in the framework of the New Széchenyi Plan. The realization of this project is supported by the European Union, and co-financed by the European Social Fund.

References

[1] Sahoo, S., Kim, S., Kim, B.I., Kraas, B., Popov, J. [2005]: *Routing optimization for Waste Management, Interfaces*, Vol.35, 24-36.

[2] Apaydin, O. and Gonullu, M.T. (2007): Route optimization for solid waste collection: Trabzon (Turkey) case study. *Global NEST Journal*, Vol. 9, No.1, 6-11.

[3] Ghose, M.K., Dikshit, A.K., Sharma, S.K. [2006]: A GIS based transportation model for solid waste disposal – a case study of Asansol Municipality, *Waste Management*, Vol.26, 1287-1293.

[4] Patent: Titrik, Á-Széchenyi István Egyetem, (2011): System for optimizing the logistic in selective waste gathering, P 11 00734.

[5] Titrik, Á (2015), Real–time based info–communication system for collecting selective waste, *Journal of Central European Green Innovation* 3 (4), 117–124.

[6] Titrik, Á (2016), Just–in–time based info–communication system for collecting selective waste, *Periodica Polytechnica Transportation Engineering*, Vol 44, No 1.

[7] Titrik, Á, Lakatos, I, Horváth, A (2015): Logistic conception for real–time based info–communication system applied in selective waste gathering, *Studia Oecologica* 9(1)., 56–67.

[8] Rosetti et al, (2002): Rossetti, R., Bordini, R., Bazzan, A., Bampi, S., Liu, R., Van Vliet, D. (2002). Using BDI agents to improve driver modelling in a commuter scenario. *Transportation Research Part* C 10, 373-398.

[9] Luenberger, D. G. (1979) Introduction to Dynamics Systems: Theory. Models and Applications, John Wiley and Sons, New York

[10] Sachkov, Y. L. (1997) On positive orthant controllability of bilinear systems in small codimensions. *SIAM Journal on Control and Optimization.* 35(1), 29-35. DOI: 10.1137/S0363012994270898

[11] Péter, T., Fazekas, S. (2014) Determination of vehicle density of inputs and outputs and model validation for the analysis of network traffic processes. *Periodica Polytechnica Transportation Engineering.* 42(1), 53-61. DOI: 10.3311/PPtr.7282

[12] Péter, T., Lakatos, I., Szauter, F. (2015) Analysis of the complex environmental impact on urban trajectories. *In: ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. Boston, Massachusetts, USA, August 2-5, 2015, DETC2015-47077, 1-7. DOI: 10.1115/DETC2015-47077

[13] Péter, T. (2012) Modeling nonlinear road traffic networks for junction control. *International Journal of Applied Mathematics and Computer Science (AMCS)*. 22(3), 723-732. DOI: 10.2478/v10006-012-0054

[14] Vadvári, T., Várlaki, P. (2015) Identification of Supply Chains Based on Input- Output Data. *Periodica Polytechnica Transportation Engineering*. 43(3), 162-167. DOI: 10.3311/PPtr.7931

[15] Dömötörfi Á., Péter T.; Szabó, K.. (2016) Mathematical Modeling of Automotive Supply Chain Networks. *Periodica Polytechnica Transportation Engineering*, [S.l.], v. 44, n. 3, p. 181-186, 2016. ISSN 1587-3811. Available at: https://pp.bme.hu/tr/article/view/9544. Date accessed: 30 oct. 2017. doi: https://doi.org/10.3311/PPtr.9544

New Clustering-based Forecasting Method for Disaggregated End-consumer Electricity Load Using Smart Grid Data

1st Peter Laurinec
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovak Republic
peter.laurinec@stuba.sk

2nd Mária Lucká
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovak Republic
maria.lucka@stuba.sk

***Abstract*—This paper presents a new method for forecasting the load of individual electricity consumers using smart grid data and clustering. The data from all consumers are used for clustering to create more suitable training sets to forecasting methods. Before clustering, time series are efficiently preprocessed by normalisation and the computation of representations of time series using a multiple linear regression model. Final centroid-based forecasts are scaled by saved normalisation parameters to create forecast for every consumer. Our method is compared with the approach that creates forecasts for every consumer separately. Evaluation and experiments were conducted on two large smart meter datasets from residences of Ireland and factories of Slovakia. The achieved results proved that our clustering-based method improves forecasting accuracy and decreases high rates of errors (maximum). It is also more scalable since it is not necessary to train the model for every consumer.**

***Index Terms*—clustering, time series analysis, electricity load forecasting, smart grid**

I. Introduction

Accurate decision making based on data-driven technologies and processes is high in demand nowadays. A large amount of data is stored in order to improve knowledge discovery and to support decision making. This also happens in the energy industry, especially by deploying smart grids. The smart grid consists of consumers (also producers) of electricity load, where every consumer is equipped with a smart meter that sends data reflecting actual electricity consumption (or production) usually in 15 or 30 minute intervals. Analysis of the whole smart grid, consisting of thousands or even millions of consumers, is important for many reasons. The most important among them are predicting and avoiding disturbances and blackouts, sustainable (environmental) usage of energy and other economic factors. The economic factors include questions about what amount of energy must be produced or saved, and also sold or bought respectively. For to make these decisions responsibly, an accurate forecasting of future electricity load values are essential. Practitioners or salespersons can be interested in forecasting of a global aggregated consumption, or an aggregated one in a small area, or even in the consumption of individual end-consumers. For some companies or producers of electricity, disaggregated end-consumer consumption forecasting is the most important task. However, this is a very difficult task because every consumer behaves differently and often unpredictably (because of a high rate of random effects). For these reasons, the end-consumers time series of electricity consumption are very noisy and often irregular. This is the reason classical forecasting methods often fail and developing new robust methods becomes important.

We propose a new clustering-based forecasting method that uses smart grid data gathered from all consumers and shares data consumption profiles to improve the accuracy of the forecast calculated for disaggregated individual end-consumer electricity consumption. Representatives of clusters are used as training data on forecasting methods to overcome noise and irregularities in disaggregated time series (avoiding also overfitting). Our method decreases the maximum forecasting errors as well.

This paper is structured as follows: Section II contains an introduction describing related works and contributions. Section III describes our approach containing the methods used for time series processing, cluster analysis, and forecasting. Section IV presents a description of the used datasets and the evaluation of performed experiments, and Section V concludes the paper.

II. Related Work

There are only a few research papers that are focused on an individual (disaggregated) consumer consumption forecasting in a smart grid. This can be due to its difficulty and the possibility of unstable results. According to our best knowledge there are no papers describing methods based on all consumers load profiles clustering to improve the accuracy of individual forecasts.

Wijaya et al. [1] use a correlation-based feature selection method for forecasting individual and also aggregate residential electricity load. They use linear regression, multilayer perceptron and support vector regression as forecasting methods. Forecasting aggregate electricity load is enhanced by clustering, but for individual consumers forecasting the

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

clustering is not used. Support vector regression and linear regression achieved best results of the forecast for an individual residential consumer, but error rate remains high around 45% of the NMAE measure (Normalized Mean Absolute Error). Ghofrani et al. [2] use spectral analysis and Kalman filtering as the method for residential customers. They evaluated three forecasting horizons: 15 minutes, 30 minutes and 1 hour. The 1 hour horizon forecasts had 30% error rate of the MAPE measure (Mean Absolute Percentage Error) on average. Koskova et al. [3] use ensembles of variable time series analysis and regression methods to produce forecasts of aggregate loads via ZIP code. The median-based weighted ensemble learning method significantly outperforms individual forecasting models.

Using clustering for improving aggregated (global) load forecasting is a more widely used method than using results of clustering for disaggregated individual consumer forecasting. Misiti [4] used discrete wavelet transform as the preprocessing of signals from consumers. The transformed data was then clustered and next optimally reclustered to a final load time series. The optimised reclustering procedure was controlled by the forecasting performance based on the cross-prediction dissimilarity index. Bandyopadhyay et al. [5] proposed context-based forecasting with clustering for improving the forecast accuracy of aggregate electricity consumption. They compared results from the proposed method with the completely aggregate method, completely disaggregate method and K-means. In our previous works [6], [7], we have focused on representations of time series and various forecasting methods to improve the forecasting accuracy of an aggregated load in the combination of K-means clustering. We concluded that the model-based representations are the best for the extracting of consumers patterns. We have proved that clustering based on different types of representations gives more accurate results than clustering using only the original load time series.

A. Contributions

According to our best knowledge, until now the combination of results of the clustering of all consumers in a smart grid and forecasting consumption of individual consumers has not been explored and evaluated. We propose a centroid-based method for forecasting that increases forecasting accuracy and decreases computational load as well. The time needed for calculations is significantly reduced from the N trained models, where N is the number of consumers, to K trained models, where K is the number of centroids (clusters created). An additional bonus is an information describing typical profiles of consumption created by clustering that can be used for further analysis.

III. Proposed Method

The description of our proposed clustering-based (or centroid-based) method for disaggregated end-consumer forecasting of electricity load can be better illustrated via a special example. We will suppose that N is a number of consumers, the length of the training set is 21 days (3 weeks) whereby in every day we will consider $24 \times 2 = 48$ measurements, and we will execute one hour ahead forecasts.

1) Starting with iteration $iter = 0$.
2) Creating of time series for each consumer of the lengths of three weeks.
3) Normalisation of each time series by z-score (keeping a mean and a standard deviation in memory for every time series).
4) Computation of representations of each time series.
5) K-means clustering of representations and an optimal number of clusters is computed.
6) The extraction of K centroids and using them as training set to any forecasting method.
7) The denormalisation of K forecasts using the stored mean and standard deviation to produce N forecasts.
8) $iter = iter + 1$. If $iter$ is divisible by 24 ($iter$ mod $24 = 0$ mod 24) then steps 4) and 5) are performed otherwise they are skipped and the stored centroids are used.

The new batch data is created by the sliding window approach that is moved each time by two values, so every one hour, because the forecasts are performed one hour (two values) ahead. Our proposed method is compared with a typical approach that trains N models and produces N forecasts. A detailed description of the methods used in the procedure above follows.

A. Normalization and Representation of Time Series

The first necessary step is the normalisation of the time series of electricity consumption by the z-score because we want to cluster similar patterns and not the time series according to the amount of energy consumption. We are using results of normalisation also for scaling (denormalisation) clustering-based forecasts. Z-score normalisation is defined as

$$\hat{x}_i = \frac{x_i - \mu}{\sigma}, \tag{1}$$

where $\hat{x}_i$ is a normalised value, x_i is an original value, $i = 1, \dots, n$, where n is length of time series data, μ is a sample mean and σ is a sample standard deviation. Z-score denormalisation is defined according to previous equation as

$$x_i = \hat{x}_i \sigma + \mu. \tag{2}$$

The next step is the computation of the time series representation, which is an input to the clustering algorithm. The modification of the time series to its representation is performed by a suitable transformation. The main reason for using representations of time series is to strength more effective and easier work with time series, depending on the application. Using time series representations is appropriate because the dimensionality reduction leads also to memory requirements reduction and to the decreasing of computational complexity. This implicitly removes noise and emphasises the essential characteristics of data. As we have shown in our previous work, model-based representations are highly appropriate for seasonal time series [6]. For a model, multiple linear regression is used for extraction of regression coefficients of two seasonalities (daily and weekly). Formally, the model can be written as follows:

$$x_t = \beta_{d1}u_{td1} + \cdots + \beta_{ds}u_{tds} + \beta_{w1}u_{tw1} + \cdots + \beta_{w6}u_{tw6} + \varepsilon_t, \tag{3}$$

for $t = 1, \ldots, n$, where x_t is the $t - th$ electricity consumption, $\beta_{d1}, \ldots, \beta_{ds}$ are regression coefficients for the daily season, s is the length of one day period ($s = 48$ in our case), $\beta_{w1}, \ldots, \beta_{w6}$ are regression coefficients for a weekly season. Weekly regression coefficients are just six, not seven, because of prevention from the singularity of the model. The $u_{td1}, \ldots, u_{tdseas}$, u_{tw1}, $\ldots$, u_{tw6} are independent binary (dummy) variables representing the sequence numbers in the regression model. They are equal to 1 in the case when they point to the $j - th$ value of the season, $j = 1, 2, \ldots, s$, in case of a daily season and $j = 1, 2, \ldots, 6$ in the case of a weekly season. The ε_t are random errors having the normal distribution of $N(0, \sigma^2)$ that are for different t mutually independent. The most widespread method for obtaining an estimate of the vector $\boldsymbol{\beta} = (\beta_{d1}, \ldots, \beta_{ds}, \beta_{w1}, \ldots, \beta_{w6})$ is the Ordinary Least Squares method [8].

B. Clustering

For clustering consumers, we used the centroid-based clustering method K-means with centroids initialization K-means++ [9]. The advantage over conventional K-means is based on carefully seeding of initial centroids, which improves the speed and accuracy of clustering and it works as follows. Let $d(x)$ denote the shortest Euclidean distance from a data point x to the closest centroid. Let us choose an initial centroid K_1 uniformly at random from the set $\boldsymbol{X}$, where $\boldsymbol{X}$ is the dataset of size $N \times n$. Choose the next center K_i, $K_i = \hat{x} \in \boldsymbol{X}$ with probability $\frac{d(\hat{x})^2}{\sum_{x \in \boldsymbol{X}} d(x)^2}$. Repeat the previous step until we have chosen all K centers. Each object from a dataset is connected with a centroid K_i that is closest to it. And after new centroids are calculated. The last two steps are repeated until the classification to clusters does not change.

In each special iteration ($iter \bmod 24 = 0 \bmod 24$) of a batch processing, we have automatically determined the optimal number of clusters to K according to the internal validation rate Davies-Bouldin index [10]. In our experiments, the optimal number of clusters ranged from 25 to 28.

After the clusters of consumers are computed, the original normalised time series are averaged according to corresponding centroids. The forecasts calculated from centroids are then denormalised according to Eq. 2 to create forecasts for every consumer.

Fig. 1 presents the clustering results of time series preprocessed by the model-based representation using the Slovak dataset. The corresponding centroid-based time series are shown in Fig. 2.

C. Forecasting Methods

Four different forecasting methods were implemented to observe the performance of the clustering-based forecasting method.

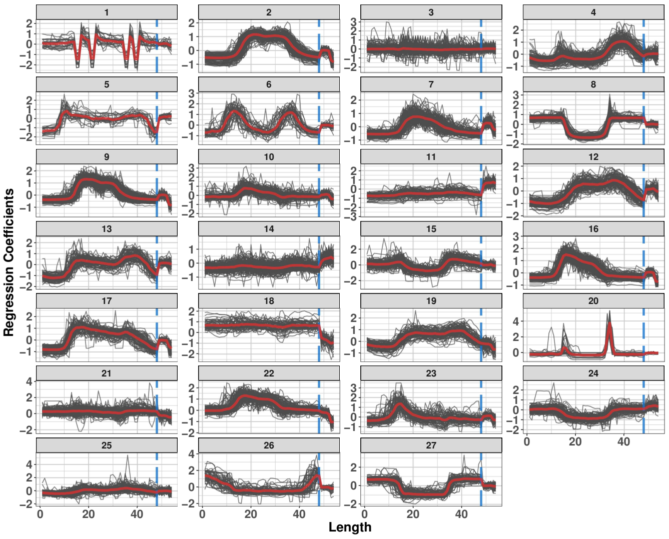


Fig. 1. 27 clusters of consumers represented by multiple regression coefficients calculated from the Slovak dataset created by K-means. Centroids are displayed by the red line. Blue dashed line splits daily and weekly seasonal regression coefficients, whereby daily (48) are placed on the left and weekly (6) are placed on the right of the blue dashed line.

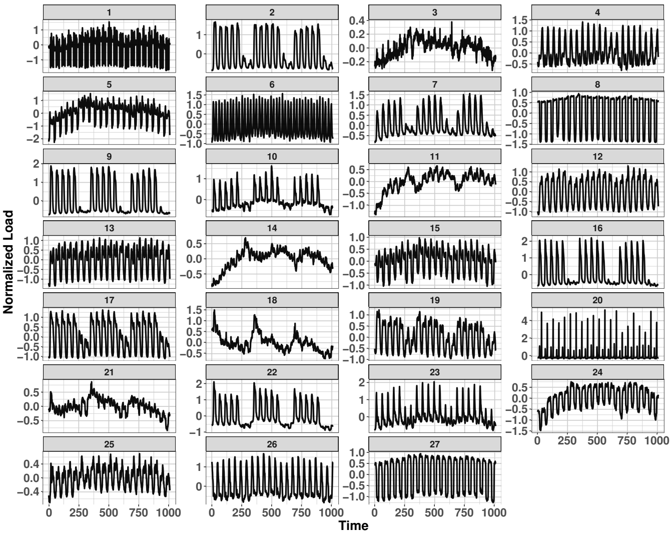


Fig. 2. Final centroid-based time series based on Fig. 1 are the input to forecasting methods.

a) SNAIVE: Seasonal naïve method is appropriate only for the time series data. All forecasts are simply set to the values of the last observations from the previous season. In our case, it means that the forecasts of all future values are set to be equal to the last observed values from the previous week.

b) MLR: Multiple Linear Regression method was also used for forecasting purposes, but the model defined in the Eq. 3 was different. In this scenario, daily and weekly attributes interacted with each other, so instead of $ds + w6$ number of attributes we have $ds \times w6$ number of interactions.

c) RF: Random Forests is an ensemble learning method that constructs a large number of trees and outputs the mean prediction of individual regression trees [11]. For adaptation to a trend change, the dependent variable (time series of electricity consumption) was detrended by STL decomposition [12] in order to improve forecasting accuracy (in Fig. 2, time series n.3, 5, 18, 19, 21, 24 and 25 shows the trend change). From the extracted trend part of the time series, future values were forecasted by automatic ARIMA procedure [13] and added to the forecast from the RF model that predicts the aggregated seasonal and remainder part of the time series. As attributes (independent variables) of the model, double seasonal Fourier terms were created. The Fourier signals are perfect for modelling seasonal time series because they consist of periodic trigonometric functions. The daily seasonal Fourier term has ds pairs of terms

$$\left(\sin\left(\frac{2\pi jt}{48}\right),\ \cos\left(\frac{2\pi jt}{48}\right)\right)_{j=1}^{ds} \tag{4}$$

and the weekly seasonal Fourier term has ws pairs of terms

$$\left(\sin\left(\frac{2\pi jt}{7}\right),\ \cos\left(\frac{2\pi jt}{7}\right)\right)_{j=1}^{ws}, \tag{5}$$

where $t = (1, \ldots, n)$. As we have experimentally verified the best setting of the number of Fourier term pairs was $ds = 2$ and $ws = 2$. The weekly seasonal component (part) from STL decomposition [12] with one day lag was also used as the attribute to the model. The hyperparameters for RF were set as follows: the number of trees was set to 1100, the minimum size of terminal nodes to 3 and the number of variables randomly sampled at each split to 3.

d) ES: Triple Exponential Smoothing is a forecasting method applied to a seasonal time series, whereby past observations are not weighted equally, but the weights decrease exponentially with time [14]. In order to adapt a model to various patterns in electricity consumption data, three different models were fitted each time and the best of them was picked to produce a forecast. These models were a full additive model with a trend component, an additive model with a damping trend and an additive model without a trend. The best model among them was chosen according to the best values of Akaike Information Criterion (AIC) [14].

We remark that only the SNAIVE forecasting method was used in both approaches. So the first approach was clustering-based (SNAIVE_Clust) and the second approach is benchmark simple one that is applied for every consumer uniquely (SNAIVE_DisAgg).

IV. Evaluation and Experiments

A. Smart Grid Data

To evaluate the performance of our clustering-based forecasting method, we used two different datasets consisting of a large number of variable patterns that were gathered from smart meters. This measurement data includes Irish and Slovak electricity load data.

The Irish data was collected by the Irish Commission for Energy Regulation (CER) and is available from the Irish Social Science Data Archive[1]. This data contains three different types of customers: residential, SMEs and others. The largest group are residential, where after removing consumers with missing data, we have 3639 residential consumers left. The frequency of data measurements was thirty minutes, so during a day 48 measurements were performed.

The Slovak data was collected within the project "International Centre of Excellence for Research of Intelligent and Secure Information-Communication Technologies and Systems". These measurements were obtained from Slovak enterprises and factories. After removing consumers with missing data, those with zero consumption and maximal consumption higher than 42 kW, the customer base comprised 3607 consumers.

The frequency of Slovak dataset measurements was every fifteen minutes, so daily 96 measurements were performed. The frequency of data measurements was aggregated to half-hourly in order to have it equal and comparable with the Irish dataset.

From each of the mentioned two datasets, a three-week testing set was chosen to investigate the performance of the implemented methods. For the Irish residential testing dataset the data measurements from 1.2.2010 to 21.2.2010 were chosen. For the Slovak factories testing dataset the data measurements from 10.2.2014 to 2.3.2014 were chosen. Moreover, we had additional 21 days (three weeks) in front of every first day of the mentioned periods. Those days were used for clustering and training the forecasting methods.

B. Evaluation Measures

The accuracy of the electricity load forecast was measured by MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) [15]. MAE is defined as

$$\frac{1}{n}\sum_{t=1}^{n}|x_t - \overline{x}_t|, \tag{6}$$

where x_t is a real consumption, $\overline{x}_t$ is the forecasted load and n is a length of data. RMSE is defined as

$$\sqrt{\frac{1}{n}\sum_{t=1}^{n}(x_t - \overline{x}_t)^2}. \tag{7}$$

C. Results of Experiments

As it was mentioned above, we have calculated one hour forecasts ahead (i.e., short-term load forecasting). For every consumer in the dataset, 504 forecasts were performed during the three weeks testing period. The statistics of results of the MAE measure are showed in the Table I and statistics of results of the RMSE measure can be found in the Table II. The statistics as the mean, median and maximum of errors were computed for every consumer separately.

On the Irish residential dataset, on average, the lowest MAE and RMSE was achieved by using our clustering-based

[1] http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

TABLE I
STATISTICS OF MAE (IN KW) OF 5 FORECASTING METHODS EVALUATED ON 2 DATASETS. IN CELLS ARE THE AVERAGE VALUES OF ALL CONSUMERS ± STAND. DEV.. DISAGG REPRESENTS COMPLETELY DISAGGREGATED CONSUMERS ELECTRICITY LOAD FORECASTING. CLUST REPRESENTS CLUSTERING-BASED FORECASTING METHOD DESCRIBED IN SECTION III. VALUES IN BOLD REPRESENT THE BEST RESULTS AMONG THE 5 METHODS.

MAE	Ireland dataset			Slovak dataset		
	Mean	Median	Max	Mean	Median	Max
SNAIVE_DisAgg	0.3807 ± 0.203	**0.1928 ± 0.147**	3.014 ± 1.3	2.6903 ± 2.854	**1.769 ± 2.27**	16.1599 ± 14.621
SNAIVE_Clust	0.3373 ± 0.178	0.235 ± 0.143	2.6605 ± 1.192	2.7873 ± 2.858	2.1479 ± 2.452	14.0958 ± 12.711
MLR_Clust	0.3403 ± 0.18	0.2394 ± 0.146	**2.6453 ± 1.187**	2.9326 ± 2.984	2.3109 ± 2.612	**14.0306 ± 12.673**
RF_Clust	0.3394 ± 0.18	0.2425 ± 0.147	2.675 ± 1.192	2.7639 ± 2.836	2.0765 ± 2.388	14.4476 ± 13.081
ES_Clust	**0.3359 ± 0.177**	0.2387 ± 0.144	2.6629 ± 1.189	**2.6752 ± 2.771**	2.0283 ± 2.357	14.1695 ± 12.816

TABLE II
STATISTICS OF RMSE (IN KW) OF 5 FORECASTING METHODS EVALUATED ON 2 DATASETS. IN CELLS ARE AVERAGE VALUES OF ALL CONSUMERS ± STAND. DEV.. DISAGG REPRESENTS COMPLETELY DISAGGREGATED CONSUMERS ELECTRICITY LOAD FORECASTING. CLUST REPRESENTS CLUSTERING-BASED FORECASTING METHOD DESCRIBED IN SECTION III. VALUES IN BOLD REPRESENT BEST RESULTS AMONG THE 5 METHODS.

RMSE	Ireland dataset			Slovak dataset		
	Mean	Median	Max	Mean	Median	Max
SNAIVE_DisAgg	0.4246 ± 0.221	**0.2147 ± 0.163**	3.2367 ± 1.371	2.8595 ± 3.006	**1.9262 ± 2.492**	16.4278 ± 14.723
SNAIVE_Clust	0.3644 ± 0.189	0.2502 ± 0.151	2.9146 ± 1.274	2.9047 ± 2.951	2.2531 ± 2.541	14.3262 ± 12.804
MLR_Clust	0.367 ± 0.19	0.2544 ± 0.154	**2.9016 ± 1.269**	3.0474 ± 3.073	2.4158 ± 2.698	**14.2595 ± 12.76**
RF_Clust	0.3664 ± 0.19	0.2572 ± 0.154	2.9301 ± 1.273	2.8832 ± 2.93	2.1812 ± 2.475	14.6839 ± 13.182
ES_Clust	**0.3628 ± 0.188**	0.2539 ± 0.152	2.9187 ± 1.271	**2.7934 ± 2.866**	2.1311 ± 2.444	14.4149 ± 12.916

forecasting method combined with exponential smoothing (ES_Clust). However, with respect to the median of errors, MLR_Clust (with MAE) and SNAIVE_DisAgg (with RMSE) methods performed best. The lowest average maximal errors had MLR_Clust method for both forecasting accuracy measures. On the Slovak dataset, on average, the lowest MAE and RMSE was achieved again by using our clustering-based forecasting method combined with exponential smoothing (ES_Clust). However, with respect to the median of errors, the completely disaggregated method SNAIVE_DisAgg (on MAE and also RMSE measure) performed best. The lowest average maximal errors had again MLR_Clust method for both forecasting accuracy measures. We can see an obvious pattern in the results of the performed experiments. The best method on average is our clustering-based method combined with exponential smoothing (ES_Clust). In most cases - three from four occasions, the completely disaggregated approach in combination with seasonal naïve forecasting method performed best with respect to the medians of errors. And finally, with respect to the maximal errors, our clustering-based method combined with multiple linear regression (MLR_Clust) had the best results. So, our clustering-based method outperformed benchmark in the meaning of average and maximal error.

A better view on reasons why our clustering-based method was mostly not better than the benchmark, with respect to the medians of errors, are shown in the following visualisations. In the Fig. 3, boxplots of hourly errors of forecasts by implemented methods for the Irish dataset are shown, while in the Fig. 4, boxplots for the Slovak dataset are shown. The reason for this situation is the fact that many zero values of errors occur. We can see that minimums and the lower quartiles of boxplots lie often on a zero value, also the median

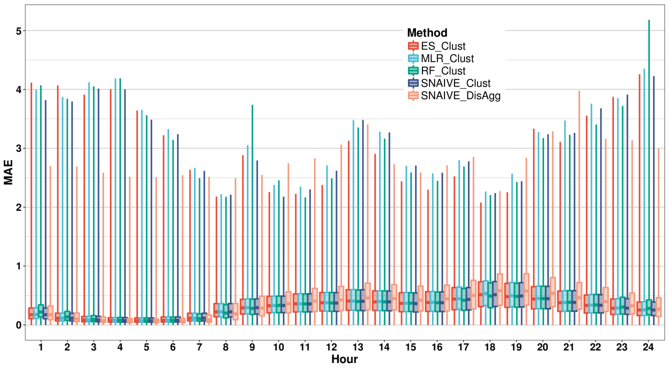


Fig. 3. Boxplots of average hourly MAE (in kW) for 5 forecasting methods on the Irish dataset. DisAgg represents completely disaggregated consumers electricity load forecasting. Clust represents clustering-based forecasting method described in Section III.

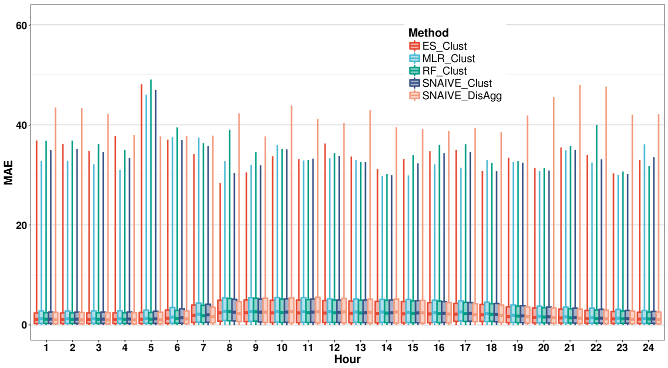


Fig. 4. Boxplots of average hourly MAE (in kW) for 5 forecasting methods on the Slovak dataset. DisAgg represents completely disaggregated consumers electricity load forecasting. Clust represents clustering-based forecasting method described in Section III.

of errors is zero for some hours. This pushes a median very low, and when forecasts are performed with the seasonal naïve method for every consumer separately, the consumption is not changing very often during the week. Therefore the use of sophisticated machine learning methods for completely disaggregated electricity consumption time series is problematic. However, for time series created as a result of clustering, it is highly recommended to use machine learning methods that for example incorporate a time series trend to a model.

V. Conclusion

In our paper, we have proposed the clustering-based forecasting method for disaggregated end-consumer electricity load using all data from consumers in a smart grid. The clustering procedure consists of the preprocessing of time series from consumers by z-score normalisation and computation of model-based representations of time series. After, the K-means based clustering with efficient K-means++ centroid initialisation is calculated and the number of clusters is optimally found by DB-index. Then the centroids of clusters are extracted and used as training sets to four implemented forecasting methods. Centroid-based forecasts are scaled by stored normalisation results (mean and standard deviation) to create a forecast for every consumer. The method is evaluated on two real datasets that consist of a large number of consumption patterns and compared with the approach that forecasts every consumer separately. We proved that our clustering-based method decreases the forecasting error in the meaning of an average on Irish dataset by 11.8% and on Slovak dataset by 0.5% with MAE measure, and the maximum (high rates of error) on Irish dataset by 13.9% and on the Slovak dataset by 15.2% with MAE measure. However, the error rates did not decrease with respect to the median because of the nature of smart meter data. In addition to the forecasting accuracy factor, our method is interesting also for its improved scalability against a fully disaggregated approach. Our method needs to train only K models (in our case about 28) instead of N models (thousands) that is leading to a huge decrease of the computational load.

More experiments to find the number of optimal clusters and the clustering method itself are suitable for a future work. Other centroid-based clustering methods like K-medians, K-medoids and Fuzzy C-means can be also used and evaluated. Handling and working with outlier or even stable consumers (who have a constant consumption profile through time) can increase forecasting accuracy as well.

Acknowledgment

This work was partially supported by the Scientific Grant Agency of The Slovak Republic, Grant No. VG 1/0752/14, and STU Grant scheme for Support of Young Researchers.

References

[1] T. K. Wijaya, S. F. R. J. Humeau, M. Vasirani, and K. Aberer, "Individual, aggregate, and cluster-based aggregate forecasting of residential demand," EPFL, Tech. Rep., 2014.

[2] M. Ghofrani, M. Hassanzadeh, M. Etezadi-Amoli, and M. S. Fadali, "Smart meter based short-term load forecasting for residential customers," *NAPS 2011 - 43rd North American Power Symposium*, pp. 13–17, 2011.

[3] G. Kosková, P. Laurinec, V. Rozinajová, A. B. Ezzeddine, M. Lucká, P. Lacko, P. Vrablecová, and P. Návrat, "Incremental ensemble learning for electricity load forecasting," *Acta Polytechnica Hungarica*, vol. 13, no. 2, pp. 97–117, 2015.

[4] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, "Optimized clusters for disaggregated elecricity load forecasting," *Revstat*, vol. 8, no. 2, pp. 105–124, 2010.

[5] S. Bandyopadhyay, T. Ganu, H. Khadilkar, and V. Arya, "Individual and Aggregate Electrical Load Forecasting: One for All and All for One," *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*, pp. 121–130, 2015. [Online]. Available: http://doi.acm.org/10.1145/2768510.2768539

[6] P. Laurinec and M. Lucká, "Comparison of representations of time series for clustering smart meter data," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2016*, oct 2016, pp. 458–463.

[7] P. Laurinec, M. Lóderer, P. Vrablecová, M. Lucká, V. Rozinajová, and A. B. Ezzeddine, "Adaptive time series forecasting of energy consumption using optimized cluster analysis," in *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 398–405.

[8] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[9] D. Arthur and S. Vassilvitskii, "K-means++: The Advantages of Careful Seeding," in *SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[10] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, April 1979.

[11] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A Seasonal-Trend Decomposition Procedure Based on Loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[13] R. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software, Articles*, vol. 27, no. 3, pp. 1–22, 2008. [Online]. Available: https://www.jstatsoft.org/v027/i03

[14] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of Forecasting*, vol. 18, no. 3, pp. 439–454, 2002.

[15] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679 – 688, 2006.

Trust in human-robot interactions

Kornélia Lazányi
Óbuda University, Keleti Faculty of Business and Management, Budapest, Hungary
lazanyi.kornelia@kgk.uni-obuda.hu

Beáta Hajdu
Óbuda University, Doctoral School of Safety Sciences, Budapest, Hungary
hajdu.bea31@gmail.com

***Abstract* — Trust within human-robot interactions (HRI) is an emerging field of organisational behaviour research. While people are by their nature mistrustful of strangers and novelties, automatons are entities on a much higher level of unknown. Their functioning and processes are hard to understand for the ordinary people, hence in HRI the role of trust is immense. In present paper the dual nature of trust - dispositional and historical – is explored through the looking glass of international literature and is researched connected to a specific subfield. The data gathered among young adults regarding autonomous vehicles points out the importance of dispositional trust, however certain factors influencing historical trust could also be identified.**

***Index Terms*—dispositional trust, human robot interaction, HRI, autonomous vehicles**

I. Introduction

Trust is an interpersonal relation, where one of the parties undertakes the risk of acting before the other person’s expected benevolent behaviour [1]. However, in the 21st century this relation has to be redefined in the light of the growing number of artificial intelligence. The topic is relevant and actual, since the aim of automation is to aid humans to increase their efficiency and productivity, however ‘mixed teams’ of humans and robots often operate sub optimally failing to reach their goal of increased efficacy. In 1997 based on their researches Parasuraman and Riley [2] stated that HRI often fail to meet the required standards due to underutilisation resulting from lack of trust in automatic systems.

The trust issues regarding autonomous systems have not been solved in the past decades [3]. What is more, with the increasing number of automated systems, the problem is more prevalent than ever [4][5]. Present paper intends to explore the attitudinal element of trust towards automation and identify its effect on the future of automated vehicles.

II. Trust

The label “trust” is often used to describe a variety of things not only in the everyday practice but in the scientific literature as well. Trust is used as a substitute variable for various phenomena, such as empathy, solidarity, reciprocity, respect, tolerance and fraternity. In general trust is a function of two agents, where actions are delivered in the light of the uncertain, yet risky future behaviour of the other party. Trust has been found inevitable for various social functions [5][6] and is the basis of a well working society [7]. As a consequence, trust is an issue tackled by many scientists. In present paper two approaches, the rational – the principal-agent theory of management sciences - and the psychological - personality trait approach of social sciences – will be tackled.

Rational approaches of trust explore the phenomenon from a self-interest point of view. Since trust decreases resources, energy and time dedicated to preparation for others’ opportunistic behaviour it decreases transaction costs. Management literature [8][9][10] investigates the issue of trust within principal agent theory and identifies control as the mean of reducing the inherent risk in such situations. Nonetheless, principal-agent theory is more complex with two human agents than in human-robot interactions (HRI). In a traditional human-to-human situation the principal (trustor) believes that the agent (trustee) will deliver (engage in certain actions), hence, he is willing to pay (do something in return) for the agent’s behaviour. While, at the very same time, the agent believes that his actions will be rewarded, hence, he is willing to deliver his actions. The situation is infused with mutual trust, or – as in most cases - the lack of it. Trust in human-to human interactions is influenced by dispositional as well as cognitive and behavioural factors [11]. According to Costa and his colleagues [12] the level of trust in a given situation is affected by the participants’ propensity to trust each other, the other party’s perceived trustworthiness, and their cooperative and control behaviours.

When the principal or the agent is lacking trust in the other party – mainly due to information asymmetry - they try to use various means of control in forms of contracts and threats of retributions. However, total control over each other’s action is not possible, not even in a panopticon (system of total surveillance) where the agent’s actions can be monitored, since his actions and inner motives are not visible for such systems [13][14]. Hence, control is only a mean of decreasing uncertainty and the inherent risk, but can never create certainty. This way, there is always place, or at least need for trust in human-to-human interactions [15]. Hosmer [16] even claimed trust to be an unavoidable connecting link in interpersonal relations.

In HRI on the other hand - owing to the preprogramed nature of the agent – principals can have (complete?) trust in the agents, since the motives and goals of the robots are clear and transparent. Hence trust issues do (should) not occur on the intentional, but on the functional level [3]. Even with the newest precision technology, super sensitive

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

sensors and extremely fast decision making due to ever increasing computational capacity, robots are (still) not flawless. They can malfunction, and the potentiality of such failures creates risk and induces mistrust in the operation of the robotic agent [17]. Control as a substitute function is easy to understand in HRI, since control means not utilising the autonomy of a robotic system to its fullest [18]. In case of insufficient trust, the scope of actions and decisions to be made by the robotic agent is limited, hence the robot is underutilised (disused). However, as trust can develop over time, operators might acquire historical trust through an array of HRI and broaden the scope of trusted robotic behaviours [5].

Social sciences, contrarily to the management approaches, often combine trust with moral duties [19]. Others regard trust as an issue of differences in personalities [20][21]. Researchers of organisational behaviour, on the other hand endeavour to explore the antecedents and nature of trust itself [22][23][24]. The research focus dates back to the ‘80s, when Frost and his colleagues [25] identified various personality related variables to have strong correlation with trust. According to Javenpaa and his colleagues [26] trust arises from the characteristics of the trustor and the trustee and their relation. Such characteristics of the trustee are competence (ability to deliver), concern (benevolence towards the trustor) and integrity. On the trustor’s side trust is a quality developed in very early childhood, the same as personality traits [27]. The propensity to trust others is assumed to be stable throughout the individual’s life, and hence it is a crucial trait that influences the trustors’ interpersonal relations and interactions [28]. This ability to trust is often labelled in international literature as dispositional trust [29]. Trust, however can be fostered and hence intensified by various factors closely related to the trustee or their interaction. According to Lee and See [30], contrarily to initial trust, which is thought to relate to affective processes, historical trust is more heavily influenced by analytic processes related to perceptions of the system and its performance. Numerous empirical studies have pointed out correlation between the performance of the system and the trustor’s level of trust [17].

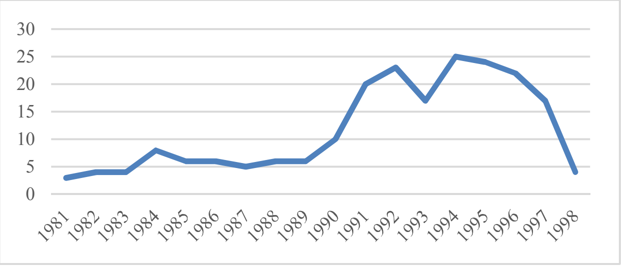
In case of HRI, the information about HOW the system is operating, WHY it is making certain decisions in certain situations and WHAT are its limitations can be one way to nurture and develop trust [31]. On top of this, historical trust may also develop through the interaction of the trustor and the trustee and the experiences related to these.

Nonetheless, the willingness to try out a system – in our case a semi-automated one – is strongly dependent on initial trust [32][5]. Information in itself – owing to the information overload of the 21st century – is not processed accurately and does not create trust when the willingness and eagerness of the trustor is non-existent. Hence, present paper approaches HRI from the point of view of initial trust and regards trust towards autonomous systems in general as a proxy for trust in autonomous systems, such as M4 (underground operated without a driver), or semi-autonomous vehicles (mostly cars) and their further automatization. The aim is to test the correlation of dispositional trust and willingness indicated by international literature, and to identify other factors that might influence the relation of trustor and trustee in HRI.

III. Research Sample and Method

In 2016, within the frame of a research program supported through the New National Excellence Program of the Hungarian Ministry of Education, a research has been initiated, where young adults (members of Generation Y) were requested to share their opinion towards autonomous systems in their lives and especially those connected to transport and traffic. The research contained 30 multiple choice questions besides those exploring the demographic data of the respondents. The respondents were found was through a targeted online questionnaire, where students of Óbuda University have been requested to forward the questionnaire to their peers living in the capital or its immediate surrounding. With the help of this snowball methodology 250 respondents could be reached, out of whom 210 have met the criteria. The distribution of respondents is displayed on Figure 1. The distribution is not a normal distribution, since the majority of students are between 19 and 24 years old, however, owing to the relative big ration (43 %) of part-time students in the sample, there are respondents of older age as well.

Figure 1: Respondents' distribution based on the year of birth



The sample consisted of 97 male and 113 female respondents. The average ages were not significantly different (24.454 with a Std. Dev.:3.791 and 23.805 with a Std. Dev.:4.525), however there was significant difference between the two genders’ travelling habits. Male respondents owned a driving licence and drove cars much more often than female respondent, who, on the other hand, tended to use public transport, which is easily accessible in the capital area explored. Further details about the independent samples’ t-test is provided in Table 1.

Table 1: Significant differences by gender

Independent Samples' t-Test	Levene's Test for Equality of Variances		t-test for Equality of Means			Male		Female	
	F	Sig.	t	df	Sig. (2-tailed)	Mean	Std. Deviation	Mean	Std. Deviation
Owning a drivers' licence	7.617	.006*	3.009	208	.003	.8433	.26650	.7257	.29544
Frequency of driving a car	.021	.884**	3.821	202.380	.000	2.237	1.427	1.487	1.409
Frequency of using public transport	7.644	.006**	-2.813	188.799	.005	2.856	1.472	3.389	1.242

*Equal variances assumed, **Equal variances not assumed

IV. Research results

When being asked directly about automatization, only 8% (N=15) of the respondents were completely rejective. However, when it came to complete autonomy of robotic systems, only 15% (N=32) were open and willing. Concerning simple automatons, such as ticket vending machines, the ratio of those preferring machines to human-to-human interaction was 69.5% to 30.5%. Half of the respondents (N=104) would trust an autonomous system to prepare food for them and only 35% to handle complex tasks (like filling the car with gasoline) instead of them. These different numbers indicate that dispositional trust might not be that important when it comes to HRI, the same is it is indicated in [5]. However, there is significant correlation between these variables, which is in line with the suggestion of dispositional trust being a personal characteristic developed in early childhood through personal experiences influenced by social and societal values (Table 2).

Table 2: Correlation of various variables describing dispositional trust

Pearson Correlation	trust automatons with complex tasks	trust automatons with simple, easily automated tasks	trust automatons with preparing your food	I am against automatisation
trust automatons with complex tasks	1	.315**	.286**	-.158*
trust automatons with simple, easily automated tasks	.315**	1	.221**	-.138
trust automatons with preparing your food	.286**	.221**	1	-.163*
I am against automatisation	-.158*	-.138	-.163*	1

*Correlation is significant at the 0.05 level **Correlation is significant at the 0.01 level

Hence the task was to identify those, who trust automatization. Interestingly trust in automatons had a gradual probabilistic nature. Those who were against automatons were almost sure not to trust them with complex tasks, but even they were ready to let them complete simple tasks that are predefined and can be controlled easily. Further details are presented in Table 3.

Hence, numbers indicate that dispositional trust is relevant when defining a person's attitude towards autonomous systems.

Table 3: Crosstabulation and probability of factors of dispositional trust

		I am against automatisation		P
		no	yes	
trust automatons with simple, easily automated tasks	no	51	8	0.61
	yes	121	7	
trust automatons with preparing food	no	86	12	0.47
	yes	86	3	
trust autonomous in public (fixed track) transport	no	102	3	0.35
	yes	70	12	
trust automatons with complex tasks	no	114	14	0.34
	yes	58	1	

As the second step, the research focused on determining factors that influence trust in semi-autonomous cars as a special subset of autonomous systems. Interestingly only week and mediocre correlations could be discovered with demographic and other personal variables. To summarise these, we can conclude that male respondents (Sig.:0.363), those of higher age (Sig.: 0.432) were more willing to trust, while those who owned a car (Sig.: 0.273), considered autonomous cars too expensive for their price (0.295) or simply trust people more than robots would not trust an autonomous car.

On top of these mediocre correlations independent samples' t-test identified two significant variables that correlated with trusting autonomous cars. As it is displayed in Table 4, those, who like to be in charge would rather not use self-driving cars, but those who thought self-driving cars and their automated responses safer would refer to travel with autonomous cars.

Table 4: Significant differences between those willing to use an autonomous car ond those who don't

Independent Samples' Test	Levene's Test for Equality of Variances		t-test for Equality of Means			No		Yes	
	F	Sig.	t	df	Sig. (2-tailed)	Mean	Std. Dev.	Mean	Std. Dev.
I like to be in charge	11.783	.001	2.014	201	.045	.849	.358	.700	.466
Autonomous cars' sensors are better than human senses, hence are safer than human driven cars	.003	0.960**	-6.309	39.345	.000	.231	.423	.767	.430

While preferring to be in charge is a personal characteristic and hence it underlines the importance of dispositional trust trusting sensors of autonomous cars indicate two things. On the one hand the individual with this perception must possess information about autonomous cars and their operation, on the other hand, his/her perception might be grounded due to past experience with similar systems. These factors are – according to international literature – basics of historical trust.

V. Conclusions

Trust within human-robot interactions (HRI) is an emerging and important field of organisational behaviour research. People are by nature mistrustful of novel things and automatons produced for the greater public are the products of the 21st century. The role of trust is enormous because the functioning and processes of robots are hard to understand for the ordinary people. For this reason, present paper introduced an inquiry delivered among 210 young adults. The results of the research were in line with the findings of international literature. While certain results emphasized the importance of dispositional trust, and a pattern of trust towards automated systems could be identified, other variables fostering historical trust could also be identified.

While for the developers and producers of autonomous systems it would be hard to influence the dispositional part of trust, since it is embedded in personal characteristics and is only influenced by social and societal interactions and can really be altered in early childhood; the historical elements are much easier to influence. Producers can foster the creation of trust by providing information about HOW their system is operating, WHY it is making certain decisions in certain situations and what are its limitations. By this simple means of mitigating information asymmetry the range of potential consumers can be easily broadened.

References

[1] Kramer, R. M., Cook, K. S. (Eds.). (2004). Trust and distrust in organizations: Dilemmas and approaches. Russell Sage Foundation.

[2] Parasuraman, R., Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. Human factors, 39(2), 230-253.

[3] Hancock, P. A., Billings, D. R., Schaefer, K. E. (2011). Can you trust your robot? Ergonomics in Design, 19(3), pp. 24-29.

[4] Dutta, R. G., Guo, X., Jin, Y. (2016). Quantifying trust in autonomous system under uncertainties. System-on-Chip Conference (SOCC), pp. 362-367.

[5] Rossi, A., Dautenhahn, K., Koay, K. L., Saunders, J. (2017). Investigating Human Perceptions of Trust in Robots for Safe HRI in Home Environments. In Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 375-376.

[6] Freel, M.S. (2003). Sectoral patterns of small firm innovation, networking and proximity. Research Policy, 751-770. 32.5.

[7] Burt, R. (2002). The Social Capital of Structural Holes. In M. Guillén, R. Collins, P. England, & M. Meyer (Eds.), New Economic Sociology, The: Developments in an Emerging Field. (pp. 148-190). Russell Sage Foundation.

[8] Eisenhardt, K. M. (1989). Agency theory: An assessment and review. Academy of management review, 14(1), 57-74.

[9] Laffont, J. J., Martimort, D. (2009). The theory of incentives: the principal-agent model. Princeton university press.

[10] Helliwell, J.F., Wang, S. (2010). Trust and well-being. National Bureau of Economic Research. No. w15911.

[11] Gill, H., Boies, K., Finegan, J. E., McNally, J. (2005). Antecedents of trust: Establishing a boundary condition for the relation between propensity to trust and intention to trust. Journal of business and psychology, 19(3), 287-302.

[12] Costa, A. C., Roe, R. A., Taillieu, T. C. B. (2001). Trust implications for performance and effectiveness. European Journal of Work & Organizational Psychology, 10(3), 225-244.

[13] Lee, J., Moray, N. (1992). Trust, control strategies and allocation of function in human-machine systems. Ergonomics, 35(10), pp. 1243-1270.

[14] Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., Beck, H. P. (2003). The role of trust in automation reliance. International Journal of Human-Computer Studies, 58(6), 697-718.

[15] Desai, M., Medvedev, M., Vázquez, M., McSheehy, S., Gadea-Omelchenko, S., Bruggeman, C., Yanco, H. (2012). Effects of changing reliability on trust of robot systems. Human-Robot Interaction (HRI), pp. 73-80.

[16] Hosmer, L. T. (1995). Trust: The connecting link between organizational theory and philosophical ethics. Academy of management Review, 20(2), 379-403.

[17] de Vries, P., Midden, C., Bouwhuis, D. (2003). The effects of errors on system trust, self-confidence, and the allocation of control in route planning. International Journal of Human-Computer Studies, 58, pp. 719–735.

[18] Chen, D., Chang, G., Sun, D., Li, J., Jia, J., & Wang, X. (2011). TRM-IoT: A trust management model based on fuzzy reputation for internet of things. Computer Science and Information Systems, 8(4), 1207-1228.

[19] Dasgupta, P., Serageldin, I. (Eds.). (2001). Social capital: a multifaceted perspective. World Bank Publications.

[20] Rukshani, K., Senthilnathan, S. (2013). A review on the relationship variables to employee morale and organizational trust.

[21] McManus, J., Mosca, J. (2015). Strategies to build trust and improve employee engagement. International Journal of Management & Information Systems (Online), 19(1), pp. 37.

[22] Dietz, G. (2011). Going back to the source: Why do people trust each other?. Journal of Trust Research, 1(2), pp. 215-222.

[23] Hardin, R. (2002). Trust and trustworthiness. Russell Sage Foundation.

[24] Jones, A. J. (2015). On the Attitude of Trust: A Formal Characterization of Trust, Distrust, and Associated Notions. In The Cognitive Foundations of Group Attitudes and Social Interaction (pp. 121-132). Springer International Publishing.

[25] Frost, T.; Stimpson, D.V.; and Maughan, M.R.C. Some correlates of trust Journal of Psychology, 99 (1978), 103-108.

[26] Jarvenpaa, S. L., Knoll, K., Leidner, D. E. (1998). Is anybody out there? Antecedents of trust in global virtual teams. Journal of management information systems, 14(4), 29-64.

[27] Lazányi, K., Bilan, J., Baimakova, K (2017): Do people in collectivist cultures trust each other more? – Comparative analysis of Hungarian and Russian students in business higher education. Journal of International Studies. In Press

[28] Salem, M., Dautenhahn, K. (2015). Evaluating trust and safety in hri: Practical issues and ethical challenges. Emerging Policy and Ethics of Human-Robot Interaction. Downloaded from: https://uhra.herts.ac.uk/bitstream/handle/2299/16336/HRI_Trust_EthicsWS_SalemDautenhahn_FINAL_2015.pdf?sequence=2 on 1st July 2017.

[29] Schaefer, K. E., Billings, D. R., Szalma, J. L., Adams, J. K., Sanders, T. L., Chen, J. Y., Hancock, P. A. (2014). A meta-analysis of factors influencing the development of trust in automation: Implications for human-robot interaction (No. ARL-TR-6984). Army Research Lab Aberdeen Proving Ground Md Human Research and Engineering Directorate.

[30] Lee J. D., See K. A. (2004). Trust in automation: Designing for appropriate reliance, Human Factors, vol. 46, pp. 50–80.

[31] Duquet, J. R., Robert, J. M., Atoyan, H. (2014). Uncertainties in complex dynamic environments. Journal d'Interaction Personne-Système, 2.

[32] Cameron, D., Aitken, J., Collins, E., Boorman, L., Fernando, S., McAree, O., Law, J. (2015). Framing factors: the importance of context and the individual in understanding trust in human-robot interaction. Downloaded from: http://eprints.whiterose.ac.uk/91238/7/cameron.pdf on 1st july 2017.

Monte Carlo Simulation in the Evaluation of the Young Functional Values

Andrzej Z. Grzybowski
Institute of Mathematics
Czestochowa University of Technology
PL-42-201 Czestochowa, Poland
andrzej.grzybowski@im.pcz.pl

Piotr Puchała
Institute of Mathematics
Czestochowa University of Technology
PL-42-201 Czestochowa, Poland
piotr.puchala@im.pcz.pl

***Abstract*—A Young measure was introduced to provide extended solutions for some non-convex problems in variational calculus. Nowadays it is very important tool applied in various optimization problems. At the core of each engineering application of the Young measure is the computation of values of so-called Young functional i.e. certain application that assigns to any Carathéodory function an integral with respect to related Young measure. However, calculating such values on the basis of its formal definition is a very difficult task which can be solved only in the most simple cases. In the paper we present some Monte Carlo method that allows one to evaluate the Young functional values in a case where the Young measure is associated with a certain class of rapidly oscillating sequences.**

Keywords—Young measure, Young functional, random numbers, Monte Carlo simulations

I. Introduction (*Heading 1*)

Solving many engineering problems often leads to many complex difficulties of mathematical or numerical nature. As an example may serve elasticity theory where there is often need to find the minimum of the energy of, say, an elastic crystal. This is done by minimizing the energy functional of the considered body. Usually this functional is of the form

$$J(f)=\int_{\Omega} W(x,f(x),\nabla f(x))dx,$$

where *Ω* is an elastic body under consideration, *f* – its displacement, *W* – the density of the internal energy of *Ω*. The problem is, that energy functionals in elasticity theory often are not quasiconvex with respect to the third variable. This is the reason why the functional *J*, although bounded from below, does not attain its infimum. Moreover, elements of any minimizing sequence of *J* are functions that rapidly oscillate (see [1] and the references cited there for a closer look). This oscillating nature of the minimizing sequences manifest itself as 'microstructure' that can be observed 'in nature', see for example [2]. It turns out that in the case of non(quasi)convex variational problems it is profitable to enlarge the space of admissible functions to the so called Young measures. This was done for the first time by Laurence Chisholm Young in [3] and his idea has since then become one of the most fruitful in investigating variational problems with no classical minimizers. The major difficulty to deal with is calculation of an explicit form of the Young measure associated with a particular minimizing sequence. Direct computations are very difficult even in relatively simple cases. This raises the need of alternative methods, such as Monte Carlo simulations to obtain their explicit form. Fortunately, the latter method has turned out to be not only very useful in practice (see [4],[5] and [6]), but also theoretically justified, see [7].

II. Some Necessary Definitions and Theoretical Results

A. A first look at the Young Measures

This section is devoted to the outline of the Young measures theory. Having in mind engineering applications we focus on a three dimensional case: the displacement *f* is a Borel mapping defined on $\Omega\subset R^3$ with values in a compact set $K\subset R^3$. The reader interested in general case $f:R^n\to R^m$ is referred to [8], where all necessary notions from functional analysis can also be found. Let Ω be a nonempty, open and bounded subset of R^3 with smooth boundary. Consider a sequence (f_k) of Borel functions on Ω with values in *K*, convergent weakly* in $L^\infty(\Omega)$ to a function *f*. Let φ be an arbitrary, but fixed continuous real valued function on R^3. It turns out that there exists a subsequence of (f_k), not relabeled, such that a sequence $(\varphi(f_k))$ of compositions is weakly* convergent not (as might be expected due to the continuity of φ) to $\varphi(f)$, but to the family $\nu=(\nu_x)_{x\in\Omega}$ of probability measures with supports contained in *K*. This family is called **Young measure** generated by the sequence (f_k). More precisely, with the above assumptions we have for any function $g\in L^1(\Omega)$:

$$\int_{\Omega} g(x)\varphi(f_k(x))dx \xrightarrow[k\to\infty]{} \int_{\Omega} g(x)\int_{R^3}\varphi(y)d\nu_x(y)dx.$$

In practice it often happens, also in important cases, that the family $\nu=(\nu_x)_{x\in\Omega}$ does not depend on *x*, it is then called a **homogeneous** Young mesure. We denote it simply by ν.

B. Young Measure realted to oscillating sequences

In order to formulate the title definition of this section we recall the notion of a Carathéodory function. Consider a function $H:\Omega\times R^3 \to R\cup\{\infty\}$. We say that H is a **Carathéodory function**, if it is Borel measurable with respect to the first variable and continuous with respect to the second one.

Definition 1

Let (f_k) be a sequence of oscillating Borel functions. We say, that a family $\nu=(\nu_x)_{x\in\Omega}$ of probability measures is a **classical Young measure generated by the sequence** (f_k), if for any Carathéodory function *H* there holds

$$\int_{\Omega} H(x, f_k(x))dx \xrightarrow[k\to\infty]{} \int_{\Omega}\int_{K} H(x,y)d\nu_x(y)dx. \quad (1)$$

We will use the abbreviation $In(f_k, H)$ for the integral on the left hand side of the above formula.

Now we will take into account a one dimensional case. Consider a function $f:[a,b)\to K\subset R$, defined on a nondegenerate interval in *R* with values in a compact *K*. We extend the function *f* periodically to the entire real line with the period $T=b-a$, and denote this extension by f^e. Given an interval $\Omega\subset R$ define a sequence (f_k) of functions according to the formula

$$f_k(x)=f^e(kx),\ x\in\Omega.$$

The sequence (f_k) will be called a Rapidly Oscillating Sequence with Uniform Representation *f*, and denoted ROSU(f). Observe that Ω - the domain of definition of the functions f_k, need not be equal to $[a,b)$ - the domain of definition of the function f. Notice that the behaviour of f_k, as k tends to infinity, is exactly the same in any neighbourhood of arbitrarily chosen $x\in\Omega$. This means, that the classical Young measure generated by (f_k) is a homogeneous one.

Recall from probability theory, that random variable *U* is said to be uniformly distributed on the interval [*a,b*), if its probability distribution has a density which equals $1/(b-a)$ on [*a,b*) vanishes outside. We will denote such random variable by $U_{[a,b)}$. We refer the reader for example to [9] for notions from probability theory.

The following proposition was stated in [6]:

Proposition 1

Classical Young measure generated by ROSU(f) is the homogoneous Young measure. Moreover, it is identical with the distribution of the random variable $Y=f(U_{[a,b)})$.

C. Young Functionals

The original idea of Young is to 'grasp' the limit oscillations of the weak* limit of the (sub)sequence of the form $(H\circ f_k)$, where f_k are rapidly oscillating functions and *H* is Carathéodory function. On the one hand, it has turned out that the parametrized measure $\nu=(\nu_x)_{x\in\Omega}$ does the job; on the other, on the basis of the Riesz representation theorem, we can look at measures as at the linear functionals defined on suitable spaces of integrands. In our case a 'suitable space of integrands' is a space $C(\overline{\Omega}\times K)$ of Carathéodory functions, which is a Banach space when supplied with the supremum norm (see [10] for details and description how to assign a Young measure to a *single* Borel function). These considerations lead us to the notion of the Young functional.

Definition 2

The **Young functional** YM is a continous linear real valued functional defined on the space $C(\overline{\Omega}\times K)$ by the formula

$$YM(H)=\int_{\Omega}\int_{K} H(x,y)d\nu_x(y)dx.$$

Thus the relation (1) of the Definition1 can be rewritten in the form

$$In(f_k,H)\xrightarrow[k\to\infty]{} YM(H). \quad (1')$$

This means that the value of the respective Young functional on H equals the limit of a sequence of integrals of rapidly oscillating functions.

III. Computing the Values of Young Functionals

Proposition 1 stated above allows one to find explicit formulae for the density function of the Young measure in a number of interesting situations. It can be achieved with the help of various probability theorems dealing with the distributions of the functions of random vectors. For example let us consider the following one-dimensional case.

Let [*a,b*) be the interval-domain of the function *f*. Let us consider an open partition of [*a,b*) into a number of open intervals $I_1,\ldots,I_n$ such that the intervals are pairwise disjoint and the union of their closures equals [*a,b*].

Let function *f* be continuously differentiable on each interval of the partition and let *f'*(*x*) ≠ 0 for all $x \in \bigcup_{k=1}^{n} I_k$

With the help of a well-known probabilistic result concerning the distributions of such functions of random variables, on the basis of the Proposition 1 one may obtain the following fact, see [6]:

Fact 1

The classical Young measure generated by any ROSU(*f*) is a homogeneous one and its density *g* with respect to the Lebesgue measure on [*a,b*) has got the following form

$$\frac{1}{b-a}\sum_{i=1}^{n}|h_i'(y)|\mathbf{1}_{D_i}(y)$$

where h_i is the inverse of *f* on the interval I_i, while $D_i = f(I_i)$ is the domain of h_i. The symbol $\mathbf{1}_\mathbf{A}$ stands for the characteristic function of the set **A**.

Example 1

Now we present an example that illustrates the usefulness of Fact 1. For this purpose let us consider the following function *f*=(*x*-1)^2 with the domain [0,2). Let the ROSU(*f*) be defined on the interval [1,4). The graphs of two exemplary elements, $f_1, f_7,$ of this sequence is presented in Fig. 1.

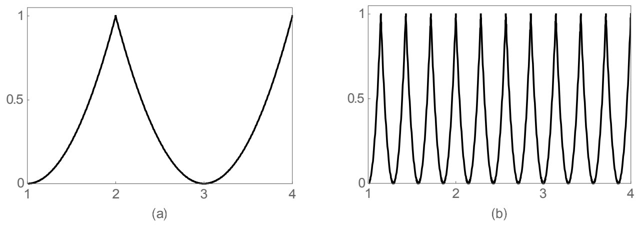


Fig. 1. Graphs of exemplary elements of ROSU(*f*) defined on [1,4), where *f*(*x*)=(*x*-1)^2 with the domain [0,2). Plot (a) shows the element f_1 while plot (b) shows f_7.

Now let us consider the exemplary Carathéodory function $H(x, y) = xy^2$ and for such an *H* let us compute the limit of the integrals $In(f_k,H)$ as *k* tends to infinity, see the left-hand side of the Eq.1. It can be seen that in the considered case the integrals $In(f_k,H)$ satisfy the following inequalities:

$$B_k \text{ -}1/k \leq In(f_k;H) \leq B_k \qquad \text{(E1)}$$

where

$$B_k = \sum_{i=[k/2]}^{2k}\left(\frac{4i}{5k^2}-\frac{2}{5k^2}\right)=\frac{2\ (1+2k\ -[k/2])(\text{-}1+2k+[k/2])}{5\ k^2}$$

with [*x*] denoting the greatest integer less than or equal to *x*. Little calculation shows that $\lim B_k = 3/2$, so by (E1) it is also the limit of $In(f_k,H)$ as *k* tends to infinity.

Now let us make use of Fact 1. On its basis we know that the Young measure related to the considered ROSU(*f*) has the probability density function *g* given by the following formula:

$$g(y)=\frac{1}{2\sqrt{y}}\mathbf{1}_{(0,1)}(y) \qquad \text{(E2)}$$

Now let us compute the value of the Young functional for the considered function *H*. We obtain:

$$YF(H)=\int_{\Omega}\int_{K}H(x,y)d\nu(y)dx=\int_{1}^{4}\int_{0}^{1}H(x,y)g(y)dydx$$

$$=\int_{1}^{4}\int_{0}^{1}\frac{xy^2}{2\sqrt{y}}dydx=\frac{3}{2}$$

So, the limit value was obtained in a much easier and direct way by the usage of the Fact 1.

Now let *H*(*x,y*)=*x*cos(*y*). Following similar calculation-steps as in the previous case we receive in this case that $In(f_k;H)$ satisfy (E1) with

$$B_k=\frac{\sqrt{2\pi}(1+2k-[k/2])(-1+2k+[k/2])C(\sqrt{2/\pi})}{k^2}$$

where *C* stands for the special function known as the Fresnel integral, i.e.

$$C(z)=\int_0^z\cos(\pi t^2/2)dt$$

Consequently, when *k* infinitely increases, the limit of B_k (and of $In(f_k;H)$ at the same time) is equal to $(15/2)(\pi/2)^{1/2}C((2/\pi)^{1/2})$, which, according to our Fact 1, should also be the value of the Young functional in this case. Indeed:

$$YF(H)=\int_{1}^{4}\int_{0}^{1}\frac{x\cos(y)}{2\sqrt{y}}dydx=\frac{15}{2}\sqrt{\frac{\pi}{2}}C(\sqrt{\frac{2}{\pi}})$$

As we see the whole knowledge about the rapid oscillations of the considered ROSU, necessary to compute the limit of $In(f_k;H)$, is "summarized" in the related Young measure, and due to our Fact 1 can be quite easily utilized to compute the Young functional value, our primal aim in various optimization tasks.

Unfortunately, there are many important situations where although inverses of the function *f* on particular subintervals do exist, they cannot be expressed explicitly. Consequently, in such cases Fact 1 is useless. However the main result stated in Proposition 1 can still be successfully utilized, because it enables one the usage of the Monte Carlo simulations in order to evaluate the values of the Young functional.

IV. Evaluation of the Young Functional Values via Monte Carlo Simulation

The idea of the Monte Carlo simulation is to draw a sample $Z_1, Z_2, \ldots Z_m$, composed of realizations of independent random variables with the same distribution as the random phenomenon under study. Based on this sample, important information concerning stochastic characteristics of the examined distribution can be derived with the help of statistical-inference tools. Indeed, by the strong law of large numbers, for any function f for which the expected value Ef(Z) exists, the arithmetic mean of the sample will almost surely converge to Ef(Z) as the sample size increases to infinity.

Monte Carlo simulations are also frequently used to evaluate values of integrals in various situations. Because we consider cases where the Young measure ν is not given explicitly, we need this approach to evaluate Young functional, i.e. the integral $YF(H)=\int_c^d\left(\int_a^b H(x,y)d\nu(y)\right)dx$. When the change of integration order is possible, then the above integral is equal to $YF(H)=\int_a^b\left(\int_c^d H(x,y)dx\right)d\nu(y)$. By Proposition 1, the Young measure ν is the probability distribution of the random variable $Y=f(U_{(a,b)})$. Consequently, we may look at the Young-functional-value as the expected value of the random variable $\int_c^d H(x,Y)dx$ and, as it was mentioned above, it can be best-estimated by its empirical mean (i.e. the arithmetic mean of its randomly generated values). Keeping this fact in mind we propose to make use of the following procedure YFE(H, f , a, b, c, d, N) to evaluate values of the Young functional:

```
Set k=1;
While k ≤ N Do Step 1 to Step 4
Step 1. Set t=Random((a,b))
Step 2. Set y=f(t)
Step 3. Set z[k]=INT(H(x,y),x,{c,d})
Step 4. Set k=k+1
Set sample=(z[1],...,z[N])
Set YF=Mean(sample)
Return YF
```

The procedure YFE is called with the following arguments: the formula H that defines the Carath´eodory function, the formula for the function f that defines the ROSU and its domain, i.e. the interval [a,b), the domain [c,d) of the ROSU itself, as it can be different from [a,b) , and the size N of the sample that will be used to evaluate the Young functional.

The subroutine Random(I) returns a pseudorandom number generated according to the uniform probability distribution defined on I. The subroutine INT(H(x,y), x,{c,d}) returns the value of the integral $\int_c^d H(x,y)dx$. Note that the integration can be done numerically, because at the moment the subroutine is called, y is a given number that was set up at the Step 2.

The usefulness of the procedure YFE will be illustrated in the following example.

Example 2

Let us consider function $f(x)=\ln(x+1)\cos(x)$ defined on the interval [0, 3π) and let ROSU(f) be defined on the same interval. The graphs of two exemplary elements of this sequence is presented in Fig. 2.

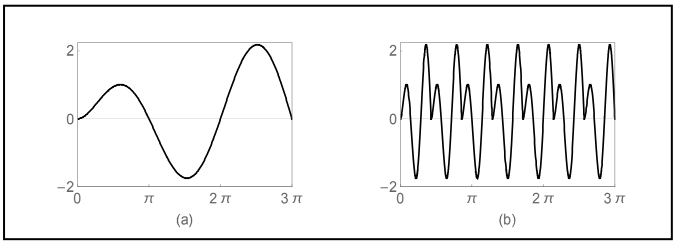


Fig. 2. Graphs of exemplary elements of ROSU(f) defined on [0,3π), where $f(x) = \ln(x)\cos(x)$ with the same domain. Plot (a) shows the element f_1 while plot (b) shows f_7

In now considered case the ROSU generating function f satisfy the assumptions of the Fact 1, but the inverses on particular monotonicity intervals cannot be expressed explicitly. Consequently, Fact 1 cannot be used to compute the Young functional values in this situation. But, as it was argued before, to evaluate these values we may adopt the Monte Carlo approach and use the procedure YFE. As an example we consider two Carathéodory integrands, the same as in Example 1. The results are presented in Table 1.

TABLE I. Results of evaluation of theYoung functional values for problems stated in Example 2.

Caratheodory integrand	*Sample size N*	*Estimated value*	*Standard Error*
$H(x,y)=xy^2$	100	65.52	6.66
	1000	63.99	2.30
	5000	64.239	0.83
	10 000	64.241	0.69
$H(x,y)=x\cos(y)$	100	19.31	2.10
	1000	18.93	0.73
	5000	19.02	0.31
	10 000	19.07	0.20

The first column of Table 1 tells us what was the Caratheodory integrand H in particular Monte Carlo experiment. In each of the experiments the task of evaluation of Young functional value $YF(H)$ with the help of the procedure YFE was performed 100 times. As a result we obtain 100 evaluations of $YF(H)$. The third column of Table 1 shows the averages of these evaluations (which can be treated as the estimated value of $YF(H)$), while the forth one contains the standard error of such an estimate. The sample size N used in YFE in various experiments is indicated in the second column.

As can be expected, the standard error of the estimated value is the less, the larger is the sample size *N*. For *N*=10000 the estimated values are rather stable; the standard error amounts to about 1% of the estimated value. However these characteristics have only statistical meaning, and it would be interesting to know what are the true errors in such experiments. To get such information one has to know the true value of the Young functional. Thus we address again the problems considered in Example 1. But now we evaluate the values *YF*(*H*) with the help of our Monte Carlo procedure. So the ROSU generating function is $f=(x-2)^2$ with the domain [0,2), while the domain of ROSU is the interval [1,4). The results of our experiments are summarized in Table 2.

TABLE II. RESULTS OF EVALUATION OF THEYOUNG FUNCTIONAL VALUES FOR PROBLEMS STATED IN EXAMPLE 1.

Caratheodory integrand	*Sample size N*	*Estimated value*	*Relative Error*
$H(x,y)=xy^2$ $YF(H)=1.5$	100	1.505	0.19789
	1000	1.500	0.00014
	5000	1.502	0.00140
	10 000	1.501	0.00070
$H(x,y)=x\cos(y)$ $YF(H)\cong 6.7839$	100	6.7869	0.00045
	1000	6.7848	0.00012
	5000	6.7849	0.00015
	10 000	6.7838	0.00002

The first column of Table 2 indicates the considered Carathéodory integrand along with the computed *true value* of Young functional, see Example 1. The second and third ones present analogous information as in Table 1, while the last one shows the average of *true relative errors* of the estimates obtained with the help of the procedure YFE . We see that the relative errors are amazingly small. Even in the case where the sample size is 1000, the relative errors are significantly less than 0.001.

V. FINAL REMARKS

The pseudocode of the procedure YFE presented in previous section can be implemented in various ways. In our research we use the following version coded in Wolfram Mathematica 10.5:

```
Mean[
 Table[
 NIntegrate[H[x,f[RandomReal[{a,b}]]],{x,c,d]]
 ,NN]
]
```

where NN is the sample size, while the remaining symbols have an analogous meaning as in the description of the YFE. In the case of Caratheodory integrands such that the function $G(y)=\int_c^d H(x,y)dx$ can be derived symbolically and it is given in an open formula for $y \in (a,b)$, then a faster version of YFE looks as follows

```
Set k=1;
While k ≤ N Do Step 1 to Step 4
Step 1. Set t=Random((a,b))
Step 2. Set y=f(t)
Step 3. Set z[k]=G(y)
Step 4. Set k=k+1
Set sample=(z[1],...,z[N])
Set YF=Mean(sample)
Return YF
```

It is this case in the exemplary problems, so in our simulations discussed in this paper we actually used the following Wolfram Mathematica code:

```
Mean[Table[G[f[RandomReal[{a,b}]]]],NN]]
```

where, for the problems discussed in Example 1 and Table 2,

$G(y)=15y^2/2$ if $H(x,y)=xy^2$ and $G(y)=15\cos(y)/2$ if $H(x,y)=x\cos(y)$. Recall that in Example 1 the integral-limits where: c=1 and d=4

We see that practical implementation of our procedure is very simple and, taking into account the accuracy of the evaluated values, it is a very competitive alternative tool for engineers, even in these cases where the Fact1 can be applied.

Finally let us note Proposition 1 can be easily generalized for the functions $f: R^k \to R^k$, *k>1,* and for such multidimensional ROSU(*f*) the Monte Carlo approach to the problem of evaluation of Young-functional-values would be much more profitable.

References

[1] P. Puchała, "Minimization of functional with integrand expressed as minimum of quasiconvex functions – general and special cases", in Calculus of Variations and PDEs, Banach Center Publications, vol. 101, T. Adamowicz, A. Kałamajska, S. Migórski and A. Ochal, Eds. Warszawa: Polish Academy of Sciences, Institute of Mathematics, 2014, pp. 168 – 186.

[2] S. Müller, "Variational methods for microstructure and phase transitions", in Calculus of Variations and Geometric Evolution Problems, Lecture Notes in Mathematics vol. 1713, S. Hildebrandt, M. Struwe, Eds. Springer Verlag, Berlin, Heidelberg, Germany, 1999, pp.85-210.

[3] L. C. Young, "Generalized curves and the existence of an attained absolute minimum in the calculus of variations", C. R. Soc. Sci. Lett. Varsovie, Classe III 30, pp 212-234, 1937.

[4] A. Z. Grzybowski, P. Puchała, "Remarks about discrete Young measures and their Monte Carlo simulations", Journal of Applied Mathematics and Computational Mechanics, vol. 14, no. 2, pp. 13-20, 2015

[5] A. Z. Grzybowski, P. Puchała, "On simulation of the Young measures – comparison of Random-Number generators", Journal of Information and Organizational Sciences, in press.

[6] A. Z. Grzybowski, P. Puchała, "On Classical Young Measures Generated by Certain Rapidly Oscillating Sequences", Proceedings of the World Congress of Engineering, San Francisco 2017, in press.

[7] A. Z. Grzybowski, P. Puchała, "On general characterization of Young measures associated with Borel functions", preprint, arXiv:1601.00206, 2016.

[8] P. Pedregal, Variational Methods in Nonlinear Elasticity, SIAM, Philadelphia, USA, 2000.

[9] T. T. Soong, Fundamentals of Probability and Statistics for Engineers, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2004.

[10] T. Roubíček, Relaxation in Optimization Theory and Variational Calculus, Walter de Gruyter, Berlin, New York, 1997.

Remarks about Inconsistency Analysis in the Pairwise Comparison Technique

Andrzej Z. Grzybowski, Tomasz Starczewski
Institute of Mathematics
Czestochowa University of Technology
Czestochowa, Poland
andrzej.grzybowski@im.pcz.pl, tomasz.starczewski@im.pcz.pl

***Abstract*—The *prioritization* of available alternatives is at the core of the multi-criteria decision making. It is typically done by assigning a *priority weight* to each of the alternatives. The weights indicate the alternatives' relative importance with respect to a given criterion. Many of prioritization methods are based on *pairwise comparisons* of the decision alternatives. As a result of such comparisons, a *pairwise comparison matrix* (PCM) that contains *priority-weights-ratios* is received. In this type of decision-making analysis a special attention is paid to so-called inconsistency analysis. It is a popular claim, that big errors in judgments about the priority ratios make the data contained in PCM useless. To "measure" the degree of PCM inconsistency various indices are proposed in literature. But the usage of these indices is justified only by some heuristics. In this paper we use simulation-based approach to the analysis of the relationships between the inconsistency indices and the errors of priority-vector's estimates. These relationships are examined under different simulation frameworks that reflect different requirements forced by the PCM-creation methodology. The simulation studies enable us to gain a deeper insight into the nature of pairwise comparisons methodology as well as new knowledge about the impact of various conventional assumptions, taken within the pairwise comparison approach, on the quality of the final priority-weights-estimates.**

***Keywords*—prioritization; pairwise comparisons; inconsistency indices; Monte Carlo simulations**

I. Introduction

Multi-criteria decision analysis is a branch of operations research that is typically concerned with decision problems with a small number of alternatives and involving a number of conflicting objectives. Nowadays it is developed within various artificial intelligence frameworks and successfully implemented in wide range of intelligent decision support systems.

Among the multi-criteria decision analysis methods perhaps the most popular is the AHP - the Analytical Hierarchy Process [1]. This method enables *decision maker* (DM) to assign priority weight to each decision alternative based on the analysis of different conflicting criteria and their relative importance. In the AHP the priorities are derived on the basis of a pairwise comparison matrix. In this decision making approach a special attention is paid to so-called inconsistency analysis. It is a common claim that serious errors in judgments about the priority ratios make the data contained in a *pairwise comparison matrix* (PCM) useless. To "measure" the degree of PCM inconsistency some characteristics, called *inconsistency indices,* are introduced in literature. However the usage of these indices is still justified only by some heuristics. In this paper we use Monte Carlo simulation experiments to gain a deeper insight into the nature of the relationships between the inconsistency indices and the errors of priority-vector's estimates. These relationships are examined under different simulation frameworks. This enables us to study the impact of various conventional assumptions, taken within the pairwise comparison methodologies, on the quality of the final priority-weights-estimates as well as their influence on the indices behavior. Consequently, the results of our study should lead to deeper understanding and more correct interpretations of the inconsistency indices.

II. Pairwise Comparisons - Notation and Basic Definitions

Let the number of alternatives in a given decision problem equals n and let $\boldsymbol{w} = (w_1,\ldots,w_n)$ denote an unknown *priority vector* (PV) i.e. a vector of the priority weights reflecting the importance of a given alternative. The pairwise comparisons of the alternatives performed by the DM result in pairwise comparison matrix $\boldsymbol{A}$, with the elements a_{ij} being the DM's *judgments* about the priority ratios w_i/w_j, $i,j=1,\ldots n$. In decision-making practice the judgments are usually expressed in linguistic terms and then transformed into an appropriate numeric scale. In such a scale the numbers indicate how many times more important or dominant one alternative is over another alternative with respect to a given criterion. In conventional AHP the most popular is Saaty's numerical scale which consists of the integers 1 to 9 and their reciprocals.

In conventional AHP the *pairwise comparison matrix* has to be *reciprocal* (RPCM), i.e. for any $i, j =1,\ldots,n$ the following condition holds:

$$a_{ij} = 1 / a_{ji} \tag{1}$$

This requirement is sometimes called the axiom of reciprocity. To meet this axiom, in AHP practice the DM fills in the upper triangle of the comparison matrix $\boldsymbol{A}$, and then for $i<j$ the elements a_{ji} are put as the reciprocals of a_{ij}.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

A given PCM is called *consistent* if it is reciprocal and its elements satisfy the condition:

$$a_{ij}\, a_{jk} = a_{ik} \tag{2}$$

for all $i, j, k = 1,\ldots,n$. It can be proved that a necessary and sufficient condition for a positive matrix $\boldsymbol{A}$ to be consistent, is an existence of a unique priority vector $\boldsymbol{w}$ satisfying $\sum w_i = 1$ and $a_{ij} = w_i/w_j$ for $i, j = 1,\ldots,n$.

It is obvious that in reality one cannot expect that the elements of PCM give exact ratios. The evaluations of the ratios may depend on personal experience, specific knowledge, the DM temporary mood and it may vary in time. Therefore in practice, even if the comparisons are done very carefully, PCM is inconsistent.

III. The issue of consistency

In reality, the PCM is usually inconsistent and, as it was argued in Introduction, we need to measure the degree of the PCM deviation from the perfect case. According to Saaty's concept the inconsistency of the data contained in the PCM is measured by an *inconsistency index* SI(n) that is computed according to the formula, [1]:

$$\mathrm{SI}(n) = (\lambda_{\max} - n)/(n - 1) \tag{3}$$

where $\lambda_{\max}$ is the principal eigenvalue of the PCM. It is well-known that for any positive *reciprocal* matrix $\boldsymbol{A}$ the value $\lambda_{\max}$ is always real, unique and not smaller than n.

Another popular inconsistency characteristic is the *geometric-mean index* (GI) that was introduced by Crawford and Williams and is defined as follows, [2]:

$$\mathrm{GI}(n) = 2[(n\text{-}1)(n\text{-}2)]^{-1}\sum_{i<j}\log^2(a_{ij}\, w_j/w_i) \tag{4}$$

The third popular inconsistency index is due to Koczkodaj [3]. To define the index we need to introduce the notion of a triad. For any three different decision alternatives we have three different priority ratios – say α, β, χ - which occupy different places in the PCM $\boldsymbol{A} = [a_{ij}]_{n\times n}$. If $\alpha = a_{ik}$, $\beta = a_{ij}$, $\chi = a_{kj}$, for some different $i, j, k \le n$, then the triple (α, β, χ) is called a *triad.* For all triads in any *consistent* PCM the equality $\beta = \alpha\chi$ holds. Equivalently in such a case both equations $1 - \beta/(\alpha\chi) = 0$ and $1 - \alpha\chi/\beta = 0$ are true. Therefore it is proposed to use the following index TI for characterization of the triad's inconsistency:

$$\mathrm{TI}(\alpha,\beta,\chi) = \min[|1 - \beta/(\alpha\chi)|,|1 - \alpha\chi/\beta|] \tag{5}$$

The Koczkodaj's inconsistency index KI of any reciprocal PCM is defined as follows:

$$\mathrm{KI} = \max(\mathrm{TI}(\alpha,\beta,\chi)) \tag{6}$$

where the maximum is taken over all possible triads in the upper triangle of the PCM.

Another inconsistency characteristic was introduced in [4]. This index is also based upon the notion of triad. Koczkodaj's index KI, is defined as the *maximum* of triad inconsistencies TI(α,β,χ), see (5). It can be seen that the value of KI is rather robust against the changes of the number of errors in the PCM if their magnitude is the same. Thus in [4] it was proposed to define a new index ATI as the *average value* of all "triad inconsistencies", namely:

$$\mathrm{ATI} = \mathrm{mean}(\mathrm{TI}(\alpha,\beta,\chi)) \tag{7}$$

where the above arithmetic mean is computed on the basis of all different triads (α,β,χ) in the upper triangle of the considered PCM.

The above-introduced inconsistency indices are used for characterization of the inconsistence of the PCM.

In multi-criteria decision analysis we are primarily interested in getting information about the *usefulness of the PCM* as a basis for PV estimation. To investigate the relationship between inconsistency indices and the quality of the PV estimates one need to clarify what prioritization method is adopted and what type of estimates error is considered. Two most popular prioritization techniques are the *right eigenvector method* (REV) introduced by Saaty [1], and the *geometric mean method* (GM) introduced in [2]. In our research we adopted both of them. However in literature (e.g [5], [6], [7]), it is reported that both methods, the REV and GM give very similar results. This fact was confirmed in the studies reported in this paper. So, similarly as in Starczewski ([8], [9]), we present here only results received while using the GM.

For measuring the errors we adopt the following loss function (an *average relative error*):

$$\mathrm{RE}(\boldsymbol{w},\boldsymbol{v}) = (1/n)\sum_i(|v_i - w_i|/v_i\,) \tag{8}$$

where $\boldsymbol{v} = (v_1,\ldots,v_n)$ is the true PV while the $\boldsymbol{w} = (w_1,\ldots,w_n)$ is its estimate received with the help of the GM method.

The average relative error RE is of special interest in the AHP. It is because even "small" errors (in terms of absolute values) may significantly change the final rankings, i.e. after analysis of the whole hierarchy, if they are big in relation to the true value.

IV. Simulations frameworks

In our Monte Carlo experiments we simulate the prioritization problems under various assumptions concerning the judgment errors. The most common and natural ones are the errors resulting from the limitations of the human brain capabilities. In literature they are usually treated as realization of random variables. In such a case the relation between the PCM elements and the true priority ratios is often expressed in the following form (e.g. [1]) :

$$a_{ij} = \varepsilon_{ij}\,(w_i\,/\,w_j) \tag{9}$$

Probability distributions (p.d.) of the perturbation factor ε_{ij} mainly involve log-normal, gamma, uniform (e.g. [10], [11],

[12], [13]) and truncated normal (e.g. [14], [15], [6], [16]), for further discussion see e.g. [17] and [8].

It is also argued (e.g. [18], [19], [20], [4]) that the errors can be additionally the results of the questioning procedure itself, erroneous entering of the data, the scaling procedure. The above types of mistakes can result in big errors in the PCM. Our Monte Carlo experiments take into account all these important situations. These errors are randomly drawn from the interval D_B with the uniform distribution $U(D_B)$.

In our studies we adopt the simulation framework described in [6], [4]. It is a generalization of the frameworks presented in [14], [15]. Within this framework the generated PCMs contain many small errors of different magnitude as well as - possibly - one large error placed at a random position in the PCM. All generated and disturbed PCMs are finally rounded to a given scale (a typical requirement in the AHP). Such simulation experiments consist of the following steps:

Step 0. Set the values of the number of alternatives n and the numbers of runs in loops N_M and N_R.

Step 1. Randomly generate n-dimensional "true priority vector" $\boldsymbol{v}=(v_1,\dots v_n)$ and related perfect PCM: $\mathbf{M}=[m_{ij}]$ with $m_{ij}=v_i/v_j$.

Step 2. Randomly choose an element $m_{i_0j_0}$ in the upper triangle of $\mathbf{M}$ and replace it with $m_{i_0j_0}\varepsilon_B$ where ε_B is a "big" error which is randomly drawn according the distribution $U(D_B)$.

Step 3. For each other element m_{ij} , $i\neq j$, $i,j\leq n,$ separately randomly choose value ε_{ij} of the small error according the p.d. π. Define *disturbed matrix* DM as matrix with non-diagonal elements $d_{ij} = m_{ij}\varepsilon_{ij}$. Diagonal elements set equal to 1.

Step 4. Define *scaled matrix* SM=$[s_{ij}]$ as matrix with elements of DM rounded to the closest value from a considered scale.

Step 5. Define *reciprocal disturbed matrix* RDM as matrix with elements $a_{ij} = d_{ij}$, and $a_{ji} = 1/d_{ij}$, $i<j\leq n$

Step 6. Define *reciprocal scaled matrix* RSM as matrix with elements $a_{ij}=s_{ij}$, and $a_{ji}=1/s_{ij}$, $i<j\leq n$

Step 7. For all matrices DM, SM, RDM, RSM compute the values of all examined indices as well as the estimates of the vector $\boldsymbol{v}$ along with these estimates' errors RE. Remember values computed in this step as one record.

Step 8. Repeat Steps 2 to 7 N_M times

Step 9. Repeat Steps 1 to 8 N_R times

Step 10. Return *all* records organized as one database.

The input for the above described simulation procedure consists of the number of decision alternatives n, the distribution of the "small errors" π, the interval D_B for the distribution of the "big" error, the numbers of loops N_M and N_R.

Let us note that in the above simulation framework by generating randomly disturbed PCMs (Steps 1-7) we in fact also "generate" *random* PV-estimates errors related to these matrices – namely RE(DM), RE(SM), RE(RDM) and RE(RSM). We want to study the distributions of these random errors and their relationship with the values of the inconsistency indices. These distributions are of our primary interest. That is because we are aware that it always may happen that a "poor" matrix results in good "estimates", and the opposite phenomena is also possible. But, as it was discussed in [4], what we are really interested in are the chances for a big estimate-error as well as the chances for small ones. So, the inconsistency indices should indicate whether such chances are "big" or "small". To study the usefulness of the indices in such a context, similarly as in [4], we analyze the relationship between their values and the *statistical characteristics* of the PV-estimates' errors distributions. For this purpose in our analysis the whole simulation database (returned in Step 10) is sorted according to the values of a given index and then split into N_C separate classes IC_i ($i = 1,\dots,N_C$). Each class IC_i is strictly connected with a unique subset of the generated PCMs – namely these which have the index values belonging to IC_i. On the other hand, each such subset of PCMs "produces" sets of random estimation errors. So the random errors are related to the class IC_i, i=1,...,N_C. Let SR_i(PCM) denotes the set of relative errors related to the class IC_i, and received when using given type of PCM (in our research the DM, SM, RDM or RSM). Note, that in case of nonreciprocal PCMs the index SI makes no sense (it may assume negative values) while the remaining indices can be naturally redefined for this kind of matrices.

To ensure some kind of objectivity, for all considered inconsistency indices we use the same computer procedure for splitting the whole range of their observed values into n separate classes. Namely, the first class is from 0 to the quantile of order $1/n$, the last n-th class starts from the quantile of order 1-1/n to infinity. All remaining n-2 classes have the same length and cover the whole interval between these two quantiles.

For each set of error values SR_i(PCM), i=1,...,N_C , we compute the arithmetic mean as well as their quantiles of the order 0.1 , 0.9 and median. One can expect that in case of a good *inconsistency* index the arithmetic mean should monotonically *increase* along with the ranges of the index values IC_i, i=1,...,N_C . The same relationship should be also expected in the case of the quantiles of the error values. Any significant violation of such a relationship would contradict the idea of the considered inconsistency-index-usefulness. In [4] it was pointed out, that the quantiles are of our special interest, because they enable one to adopt well-established statistical inference theory in the PCM acceptance approach. That is why good indices should be correct in indicating the changes in the quantiles values, or equivalently, the changes in probabilities of "small" and/or "big" estimation errors.

V. Results

The results of the simulation experiments were obtained within framework where the probability distribution of the small error (compare Step 3) is one of the following four most frequently considered in literature types of p.d.: gamma, log-normal, truncated normal, and uniform one. In each case the parameters of the distributions are set in such a way that their expected values equal 1. The support of the truncated normal and the uniform distribution is the interval $D_S = [0.5, 1.5]$. In case of the remaining two distributions, their parameters were prescribed in a way ensuring that the probability of the interval D_S is greater than 0.98. We assume the big error interval is $D_B = [2, 3]$ and the numbers of runs in our simulations were $N_M = 12$, $N_R = 500$. To compute the required statistical characteristics, the values of indices were split into N_C□ =15□ classes.□In Table 1 we present values of the Spearman rank correlation coefficients between the statistical characteristics chosen as discussed in Section 4. In order to eliminate the impact of outliers (i.e. the extremely small or extremely big judgment errors) on the relationship, while computing the Spearman coefficient two classes (the first one and the last one) were omitted. We have already mentioned that the Saaty's index is uninterpretable in cases of matrices of types DM and SM, so the values of its characteristics can be ignored for these types of matrices. However they were computed anyway, so we put them into Table 1 just to let the reader know how the index behaves for nonreciprocal matrices.

Spearman correlation coefficient is the most important characteristic of the investigated relationship, because it shows the "degree of accordance" between increments of indices and increments of related estimation-errors, expressed by the magnitude of statistics analyzed in Table 1. As we already argued, it is essential for any "good" inconsistency characteristics that their Spearman rank coefficient with all the quantiles and means should be positive and as close to 1 as possible. Bearing this in mind we see that the results presented in Table 1 reveal some important phenomena. Firstly, the best correspondence between all the indices and the magnitude of the estimation errors is for the matrices DM, i.e. nonreciprocal matrices without scale. Secondly, all indices perform poorer for reciprocal matrices (in cases of both types the RSM and RDM). At this moment we have to emphasize that the only reason justifying the axiom of forced reciprocity in AHP is that it enables one to use the SI, but even in this case this index works poorer than ATI or even GM. Taking it into account makes the axiom of reciprocity as well as the usage of SI questionable. Next observation is that the index ATI turns out to be the best indicator of the PV estimation quality when the PCM is nonreciprocal. Quite surprising fact revealed in our studies is that for reciprocal PCMs without adopted scale all indices *have no relation* with the quality of PV estimates. On the other hand, in view of our studies, when the inference is based on the SRM (i.e. on both reciprocal and scaled matrices) then the indices are correctly correlated with the mean of estimation errors, and again the best rank correlation is manifested by the ATI.

VI. Final remarks

Very similar results were obtained for greater number of alternatives. We perform our studies for $n = 4, 5, 6$. In each case we observe the same phenomena as described above. In our studies we analyze situations where the "big error" is present. Let us stress, that in the context of the study of the indices performance, these situations related to big-error-existence are the most important cases. That is because the indices are often used in a number of procedures leading to the PCM's quality improvement (see e.g. [5], [21]) and such an improvement is necessary only when the big error is present. Thus it is very important that they should properly indicate the changes in "inconsistency" in such cases.

TABLE I. Average Spearman rank correlation coefficients between the mean values of the inconsistency index computed in its classes IC_I ($I = 1,...,15$) and the indicated statistical characteristics of PV estimation errors corresponding to these classes. The presented results are computed on the basis of 96 000 PCMs randomly generated according the simulation procedure described in Section 4. The number of alternatives $N = 4$.

RE(DM)		**Correlations with the errors RE(DM)**			
		SI	***GI***	***KI***	***ATI***
Mean		-0.65	0.98	0.93	1.00
Quantiles	$p = 0.1$	-0.06	0.53	0.71	0.94
	Median	-0.45	0.97	0.82	1.00
	$p = 0.9$	-0.96	1.00	0.98	0.99
RE(SM)		**Correlations with the errors RE(SM)**			
		SI	***GI***	***KI***	***ATI***
Mean		0.31	0.65	0.72	0.73
Quantiles	$p = 0.1$	-0.33	0.99	0.34	0.98
	Median	-0.77	0.99	0.74	0.99
	$p = 0.9$	-0.20	0.64	0.64	0.74
RE(RDM)		**Correlations with the errors RE(RDM)**			
		SI	***GI***	***KI***	***ATI***
Mean		-0.53	-0.53	-0.31	-0.54
Quantiles	$p = 0.1$	0.34	0.28	-0.45	-0.49
	Median	-0.06	-0.07	0.37	0.31
	$p = 0.9$	-0.93	-0.93	-0.63	-0.89
RE(RSM)		**Correlations with the errors RE(RSM)**			
		SI	***GI***	***KI***	***ATI***
Mean		0.52	0.57	0.19	0.71
Quantiles	$p = 0.1$	-0.03	0.11	-0.15	-0.31
	Median	-0.68	-0.68	-0.24	-0.17
	$p = 0.9$	0.43	0.35	0.10	0.30

In the cases where big errors are rare all indices perform better. Especially the KI performs much better than it is reflected in Table 1, if there are no big errors within the simulation framework (i.e. the big error parameter equals 1). But this fact is easy to explain because in such a case the KI perform similarly as ATI - the lack of big errors makes the values of both indices close one to another. When the big error appears in no more than 30% of the PCMs then all indices have positive values of the Spearman rank coefficient for all considered types of matrices. In such cases the results are similar to those presented in [4]. Nonetheless the negative impact of the forced reciprocity and the assumed scale is observed in every simulation framework and for any examined number of alternatives. On the other hand one should be aware that adoption of some scale is unavoidable. However from our research it results that the scale should be much richer than the Saaty's one to enable more precise expression of the DM judgments.

References

[1] T.L. Saaty, "The Analytic Hierarchy Process," McGraw Hill, New York (1980).

[2] G. Crawford and C.A. Williams, "A note on the analysis of subjective judgment matrices," J. Math. Psychol. 29, 387-405, (1985).

[3] W.W. Koczkodaj, "A new definition of consistency of pairwise comparisons," Mathematical and Computer Modelling 18(7),79–84, (1993).

[4] A.Z. Grzybowski, "New results on inconsistency indices and their relationship with the quality of priority vector estimation," Expert Systems With Applications 43, 197–212, ISSN: 0957-4174 (2016).

[5] M. Lamata, P. Peláez, "A method for improving the consistency of judgements," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (6) 677-686, (2002).

[6] A.Z. Grzybowski, "Note on a new optimization based approach for estimating priority weights and related consistency index," Expert Systems with Applications 39 , 11699–11708 (2012).

[7] C. Lee, H. Lee, H. Seol and Y. Park, "Evaluation of new service concepts using rough set theory and group analytic hierarchy process," Expert Systems with Applications 39, 3404–3412 (2012).

[8] T. Starczewski, "Relationship between priority ratios disturbances and priority estimation errors," Journal of Applied Mathematics and Computational Mechanics, vol. 15, no. 3, pp. 143-154, (2016).

[9] T. Starczewski, "Remarks on the impact of the adopted scale on the priority estimation quality," Journal of Applied Mathematics and Computational Mechanics, vol. 16, no. 3, pp. 105-116, 2017.

[10] D.V. Budescu, R. Zwick and A. Rapoport, "Comparison of the analytic hierarchy process and the geometric mean procedure for ratio scaling," Appl. Psychol. Meas., 10, 69-78, (1986).

[11] F. Zahedi, "A simulation study of estimation methods in the analytic hierarchy process," Socio-Econ. Plann. Sci. 20, 347–354, (1986).

[12] I. Basak, "Comparison of statistical procedures in analytic hierarchy process using a ranking test," Math. Comp. Model. 28, 105–118, (1998).

[13] A.Z. Grzybowski, "Goal programming approach for deriving priority vectors - some new ideas," Sci. Res. Inst. Math. Comp. Sci., 1(9) 17-27, (2010).

[14] E.U. Choo and W.C. Wedley, "A common framework for deriving preference values from pairwise comparison matrices," Comp. Oper. Res. 31, 893–908, (2004).

[15] C-C. Lin, "A Revised Framework for Deriving Preference Values from Pairwise Comparison Matrices," Euro. J. Oper. Res. 176, 1145-1150, (2007).

[16] P. Kazibudzki, "Redefinition of triad's inconsistency and its impact on the consistency measurement of pairwise comparison matrix," Journal of Applied Mathematics and Computational Mechanics, vol. 15, no. 1, pp. 71-78, (2016).

[17] T.K. Dijkstra , "On the extraction of weights from pairwise comparison matrices," Central European Journal of Operations Research, 21, 103–123, (2013).

[18] E. Bulut, O. Duru, T. Keçeci and S. Yoshida, "Use of consistency index, expert prioritization and direct numerical inputs for generic fuzzy-AHP modeling: A process model for shipping asset management," Expert Systems with Applications, 39,1911–1923, (2012).

[19] S. Lipovetsky and M.M. Conklin, "Robust estimation of priorities in the AHP," European Journal of Operational Research 137, 110–122 (2002).

[20] J. Temesi, "Pairwise comparison matrices and the error-free property of the decision maker," Central European Journal of Operations Research, 19, 239–249, (2011).

[21] W.W. Koczkodaj and S.J. Szarek, "On distance-based inconsistency reduction algorithms for pairwise comparisons," Logic Journal of the IGPL, 18(6), 859-869, (2010).

Automatic assessment of assignments for Android application programming courses

Matej Madeja, Jaroslav Porubän
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia

Abstract—**This paper presents a solution of creating a testing environment for Android applications in programming courses. Appropriate testing methods and suggestions for the testing environment are consulted by the authors with a mobile application development company. The paper also analyzes basic student mistakes, looks for solutions to their detection by automated testing, and suggests appropriate test tools on their basis. At the same time, the paper contains an assessment and testing experience for particular tools. In addition, it compares the performance of emulator and real device tests, and the proposed tools are partially tested in particular course.**

I. Introduction

Mary Meeker's report [1] from May 2017 shows that on average American adult spent 3.1 hours per day on a mobile device in 2016 and this number is still increasing. The report also points to the fact that since 2011 Android is the most used mobile operating system and from April 2017 has the major market share among all operating systems in the world [2]. Due to the above popularity a research [3] with 400+ IT professionals has been conducted and 76% of them stated they develop native Android applications and companies in which they are employed produce mostly applications for Android platform. This is the reason why MOOC (massive open online course) are so popular in this business (see [4], [5], [6]).

Many IT companies in the area of our university demand experienced programmers [7]. Responding to this situation, our university launched the course *Application Development for Smart Devices* in 2015 with a major focus on the Android platform. One of the inconvenient tasks in programming courses is to evaluate the students' assignments. In the standard procedure, the teacher must run each program, enter test data and evaluate the correctness of the solution. As the course is usually attended by tens or hundreds of students, therefore, the evaluation of the assignments is very time-consuming.

Until today 2 runs of the course were conducted, where the assignments were evaluated in a typical way - by the teacher. It was clear that it is not possible to evaluate the students' solutions properly and objectively. There is no way to test all possible inputs, so the question arises: How to test such applications? Furthermore, when assignment is submitted by a student, it is often seen that the student did not do the job by himself. In the case of typical evaluation of assignment is this fact really difficult to prove. Automated testing can help us solve the problem of student's assignment originality through an automatic plagiarism detection (more in [8] and [9]). Test-driven development (students create implementation according to tests) also shows that it is beneficial for the quality of programs [10], so automatic student's solution testing should also improve their quality of program functionality (probably source code, too).

Related research, with focus on related surveys of mobile application testing and teaching, is presented in Section II. Later in Section III we briefly describe an application for which testing tools will be suggested. In Section IV we analyse the most common mistakes of students, Section V is dedicated to the design of the testing environment and partial results are described in Section VI. Expectations and plans for the future are discussed in Section VII and conclusions in Section VIII.

II. Related work

Testing mobile applications is common, but not with a focus on automatic assessment of assignments in programming courses. We first searched for existing automated testing solutions for students' assignments in MOOC for Android programming, but none of the courses found provide assessment of student assignments by automated testing. These courses only deal with application development itself and mostly rely on manual app testing (either by teachers or directly by student in his or her own interest). Although some courses contain automated tests development in the curriculum (e.g. [11], [12], [13]), but again, none of the courses validate students' solutions by automated system.

Unfortunately, we did not find any research that addresses the same issue. We also sought inspiration at prestigious universities such as Stanford University, MIT[1], or Harvard University, but there was nothing to suggest from the general information that they use automated tests in similar courses and we have no access to their internal system. Nevertheless, there are a number of practical tools that could be explored in terms of target use.

Evidence that the topic of testing (of mobile apps) is generally up to date is that many developers write about it on their blogs and business pages. Already in 2007, Pecinovský [14] claimed that students should be taught by the step-by-step

[1] Massachusetts Institute of Technology

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

coding method. Ten years ago, in 2017 DZone team issued a guide called *Automated Testing* [15] in which they claim that there is no longer any other way to develop software only through Continuous Delivery or Integration. Automated testing is essential to DevOps and Continuous Delivery and in order to integrate continuous testing effectively into a DevOps toolchain, the following essential features are key to evaluating an automated testing platform:

- Support for a variety of languages, tools, and frameworks.
- Cloud testing.
- The ability to scale rapidly.
- Highly automated.
- Security.

Of the over 400 IT professionals questioned in DZone's research, most said they generally use integration, component or performance testing in Continuous Delivery. Furthermore, Sauce Labs issued ebook *Mobile app testing: Main challenges, different approaches, one solution* [16] in 2017 in which it mentions that the biggest challenges for automated testing applications are:

- time saving,
- cost reduction,
- repeatability of tests,
- increased coverage of app features,
- re-usability.

Nevertheless, they (and other researches, e.g. [17], [18]) claim that manual testing for mobile apps is still unnecessary for some device-specifc function, such as location data or other environmental sensor data. The ebook also provides an overview of testing tools and solutions for testing infrastructure. Since Android is the most popular mobile OS (2011)[2], many other researches for automated testing has been done, such as [19], [20], [21].

In 2016 Wilcox described in paper *Testing Strategies for Automated Grading of Student Programs* [22] a set of strategies for testing students' programs that are more effective than regression testing at providing detailed and relevant feedback to students. In addition, he discussed some of the issues that arise in the context of automated grading and their solutions, such as grading performance, non-terminating programs and security issues. In recent years there are many other papers, e.g. [23], [24], [25], in which authors look at the automatic assessment of assignments from different perspectives.

III. Application Makacs

In the first two runs of the course *Application Development for Smart Devices* (2015, 2016), the main task was to create a sports monitoring and tracking application called *Makacs*. The implementation of this application was designed to bring the student into many topics in the development of Android applications. The application had the following requirements:

- min. 5 activities, of which at least 1 will include calorie counting,
- min. 1 service (recommended for counters - duration, pace, distance, calories),
- min. 1 broadcast receiver,
- min. 1 activity with own list implementation,
- data persistence with *SQLite*[2],
- use of min. 1 sensor,
- communication with an external web service through the API,
- min. 2 languages variants,
- min. 1 custom extension.

The app have to be targeted for Android devices with API 19+ and did not have any other implementation restrictions, it only had to meet the requirements and features had to be realistically usable. The creativity and the uniqueness of the implementation were left to the student, while the originality of the solution was assessed, of course, by teacher's subjective opinion. Because we have decided to find a solution for automatic assessment of students' assignments students will have to be restricted to a greater extent (due to the testing environment), which is a risk that students will be less creative.

IV. Most common students' mistakes

Based on the Section II, we conclude that there are many solutions and insights into test systems (or environments) in the learning process, but none of mentioned is used to test mobile applications. Our first question, then, was: What to test in student solutions? In generally the most common Android app failures are the following [26]:

1) Application failure when installing.
2) Application crash during execution.
3) Problems with scaling or deploying elements on the screen.
4) Non-responding application from unavailable sources.
5) Problems of viewing content in landscape or portrait mode.

These were all common issues for apps that people hate the most. Practical solutions in Section II show mobile applications are tested, but the testing of students' programs may lead to unique situations which, in practice, do not need to be treated to such an extent. Due to the possibility of occurrence such unique situations, we have been consulting the most often mistakes of new Android app developers with mobile development IT company *Wirecard*. After collecting the recommendations, we first selected 4 random student apps from the course in 2015 and 2016 and tested them manually. Selected applications have encountered issues such as navigation problems, data loss on activity restart (device rotation or change of system language), and some logical issues (such as device uptime dependency for counting duration of sport activity).

Since app errors were approximately distributed in the same number, we decided to test another 4 apps to get more accurate results. After trying them some new issues appeared such as the inability to install the app and forgotten service at activity restart, caused by change of system language. In generally, other issues were only repeated. Because we wanted to make

[2] https://www.sqlite.org/

TABLE I
MOST COMMON STUDENT'S ISSUES IN MOBILE DEVELOPMENT.

Issue	Failed Applications
Navigation & UX	7
Data loss on activity restart (e.g. device rotation)	6
Logical problems (e.g. usage of device variable data)	3
Unable to start or install app	2
Forgotten services at activity restart	1

sure we will not find other issues yet, we tried to test two more apps where there was no new issue found. Together, 10 applications were tested and the test results are shown in the Table I.

Manual testing was carried out on 2 facilities:

1) Nexus 5, API 25, Android 7.1.1, emulator.
2) Prestigio 5453 DUO, API 19, Android 4.4.2, real device.

Based on these results, we would like to concentrate mainly on these issues in test cases for our automated testing environment.

V. TESTING ENVIRONMENT

Now we know the most common mistakes of students and applications and the next question arise: How we can build a test environment to detect these issues/errors? The differences between testing of desktop programs and mobile applications, as well as problems arising from the testing of mobile applications, are described in more detail in Wasserman's article *Software engineering issues for mobile application development* [27]. Some of these issues and new problems found by us are discussed below.

A. Static testing

The plan for the course curriculum is to have 4 deadlines to submit partial assignment solution during the semester, which will be tested in testing environment. At the beginning of the course it is a big problem to motivate students to work. Lectures compared to seminars are usually not at the same time level, meaning that students often lack the theoretical knowledge to start programming. However, when designing mobile applications, students can design a UI for their application and become familiar with UI components in the IDE, where the work rests solely on editing XML documents.

We have devised the task at the beginning seminars of the course, which will be able to test the testing environment. We have created a task where the students have to create their own application design and prepare it in IDE. Created (or generated) XML files can then be uploaded and the test platform can check them. The seminar documents describe exactly what elements must contain specific activity resource file (file defining UI elements) and what identifiers must be defined in it. The test system then tests only the uniqueness of the identifiers for every activity file and whether the identifier is associated with the correct element. This automated test layer (we call it *xmlchecker*) has a static character (static tests) and requires an exact directory structure with resource files. On the other hand, in order to check whether the students actually created the real-world design of the application, we require that screens of activities be sent and then the teacher check them manually (so that identifiers can not be fake-generated and sent for evaluation). This type of manual checking is not very time-consuming and we have judged it to be sufficient.

B. Test pyramid and tools selection

The other 3 submissions contain, in addition to XML checks, real test of the functionality of the students' source code, so in the following testing we will use several tools. The question arises how the test structure should look like. Typically, test cases are divided into several layers. For Android apps, the following test layers are common [26]:

- unit tests,
- integration tests,
- operational tests (also called functional or acceptation),
- system tests.

In our case, we chose to use the concept of the *test pyramid* [28]. The concept includes the following 3 tiers:

- unit tests,
- integration tests,
- UI tests (or system tests).

When testing student assignments we do not need a separate layer for operational tests, because they will be included in UI tests (according to existing testing tools). The test pyramid concept says any automated testing strategy should have more low-level unit tests than high-level UI tests. From a viewpoint of the complexity of performing tests, unit tests are the easiest and fastest, on the contrary, performing UI tests takes a relatively long time. By using these tiers in our test platform, where hundreds of applications are tested and the idea of this concept is preserved, we can accelerate the testing process and increase test reliability (by testing the same or similar functionality by various tools from various tiers).

For listed test tiers of test pyramid we compared the various open-source test tools so that we can test the most functionality of student applications automated. We also consulted with *Wirecard* company. For unit testing, Google recommend to use *JUnit* with *Mockito* [29] tool to create mock objects (needed for tests where application context is needed). The purpose of unit test is to isolate the smallest source code units and check, even in isolation, they work correctly. In Android applications are often even the smallest tasks depended on the context of the app, in which case *Mockito* can help us.

For integration tests, *Wirecard* advised us to use *Robolectric*[3] framework with whom they had a wealth of experience. Due to the close offer of similar tools, we have not found a suitable competitor, so we have decided to use this tool.

As for the last layer of UX testing, choosing a test tool was not as simple as the previous ones. Because there is a lot of UI testing tools and they overtake each other in their features, we have decided to do our own comparison [30]

[3] http://robolectric.org/

of these tools in 2017. We have looked at this tools from a practical point of view, and in particular from the point of view of their capability to test. The most popular tools were *Appium*[4], *Robotium*[5] and *Espresso*[6]. Though, the most important indicator for us was their ability to cover the widest possible range of different test cases. We looked at factors like app support type (native, hybrid), context (device, application), IDE intergration, support of OS vendor, emulator vs. real device testing, etc. The best tools from this point of view were *Espresso*, *Appium* and *Robotium*. *Espresso* has the best overall score, so we decided to use it in our solution. It has the ability to record test cases, which greatly facilitates the creation of tests. In the UI tests, we also decided to use the *Monkey*[7] tool to perform stress tests (different and fast gestures above the app UI).

C. Writing tests

Because the proposal was not yet implemented in the course, we created a sample *Makacs* application that we experimented with. The goal of the test pyramid is appropriate for the assessment of student's assignments because unit tests could be done very quickly. First we started writing unit tests to get them as much as possible. In the sample app proposal, a method of calculating calories, change the units of distance (km, mi), weight (kg, lb), etc., was precisely defined. Nevertheless, after a short while, we found out that we were actually tested everything possible. Together, 6 different unit tests were made with a size of 5-16 lines. And there was a problem because the notion that UI tests would be even smaller was unrealistic. Nevertheless, we continued the chosen concept and tools.

Some tests need to be developed in conjunction with the *Android SDK*, but we do not want to deploy the application to virtual or real device, most often because it is time-consuming. In that case, we used the *Robolectric* framework, as a part of integration tests, that mocks the *Android SDK* and thus eliminates the `RuntimeException` exceptions resulting from the empty, so-called "stub" implementation of methods. Tests are running directly in JVM, so we did not need a virtual or real device yet. This tool is actually a headless UI test framework (possible to call action on UI element). When writing tests, we found that UI headless testing errors were detected, pointing to some problems in the test framework. We found out that problems occur with every new version of the *Android SDK* when a new implementation of stub methods is needed (mocking of *Android SDK*). Therefore, it is not a full replacement for UI testing tools because UI testing tools do not have such dependencies. The best choice is a combination of both approaches (headless and classic UI testing). However, this headless UI test has not been reliable due to the various UI elements in the various Android APIs and the resulting conflicts.

A great feature of *Robotium* was ability to automatically clear the test environment for each test (such as app settings). Problems, however, occurred when running multiple database access across tests because tests fell due to singleton implementation of the `DatabaseOpenHelper` class (due to usage *ORMLite*[8]). After each test performed, it is necessary to release the singleton manually.

At the same time, question arose how to test intents of individual activities or starting/stopping app services. From the student application, the desired action (new activity, system application request, etc.) is expected to be executed after clicking on a specific button (element ID). *Robotium* provides a great and simple solution for tracking intentions. Likewise, for services, it provides a stack of all existing services in the application. Implementation is more complicated during testing, because it is always necessary to erase the stack to test a particular service. Together, 11 tests for the *Robotium* tool were performed in a range of 2-10 rows. All unit and integration tests are able to run using the *gradle*[9] tool.

As part of the integration tests, we knew that after a specific action the service with that name started or stopped. However, we was not able to test the functionality of the service, which was a major part of the *Makacs* logic (calorie counting, distance, etc.). Despite the fact that *Robolectic* allow testing of service functionality, the solution was hard to implement. We found the *AndroidJUnit4*[10] tool, which is much easier to implement using a binder, on the other hand this tool needs an Android device (virtual or real). Together we created 6 tests for the main service of the 24-44 line size.

As the last and highest layer of test environment was UI testing. *Espresso* is a young test framework, which is also felt by the developer because the writing of the tests is very intuitive. Even general tests (such as navigation) can be created using a test recorder, which greatly accelerates the creation of tests. The only downside of recording is that it does not look for component identifiers but for their texts and order in a particular activity. This approach is not accurate, so you need to set the identifiers manually. However, it's faster than writing all tests manually.

The problem occurred when running multiple tests at once. *Espresso* remembers the status of the application's database as tests run on a specific device. This is undesirable in some cases (especially when modeling specific use case, e.g. data recovery from database of crashed app), so it is necessary to delete the database manually in these cases. Tests are triggered at random, so it is necessary to individually check before each test whether the application is in the required state to start particular test.

As part of the UI testing, we used the stress testing tool *Monkey*, which runs random 10,000 random gestures over each student solution to get the application into an unexpected error.

[4] http://appium.io/

[5] https://robotium.com/

[6] https://developer.android.com/training/testing/espresso/

[7] https://developer.android.com/studio/test/monkey.html

[8] http://ormlite.com/

[9] https://gradle.org/

[10] https://developer.android.com/training/testing/unit-testing/instrumented-unit-tests.html

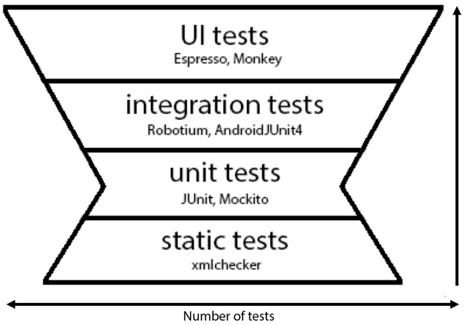


Fig. 1. Tiers of proposed testing environment.

Figure 1 shows the actual proposed structure of the test environment. As we can see, the testing enviroment in terms of the number of tests is the opposite of the test pyramid concept. This is due to the fact that if we do not want to limit the students' solution to a large extent in the assessment of students' assignments, we need to focus on testing that is not dependent on implementation details (at most IDs of UI components). We have also added static tests to the test process for tasks that a student can accomplish without the need of programming on a given platform. This result does not say that the concept of the test pyramid is poorly designed, but for the testing of students' assignments, in our case, we had a different character from the view of tests number.

VI. Partial experimental results

All these suggestions were prepared for the fall 2017 semester of the course. In order to know the assumption that the testing environment is capable of evaluating assignments, we performed a number of possible tests on a random sample. Some results of manual testing have been already described in Section IV, because based on these results, we focused on selecting test tools and implementing particular test cases.

From the proposed test tools, student solutions from 2015 and 2016 could be tested only by stress testing by the *Monkey* tool, which is not dependent on the specific implementation of the solution. We tested 10 randomly selected applications that were approved by manual teacher testing, with 50% of testing applications failing. In the test 10,000 random gestures were used over the application, and the most common failures were `NullPointerException`, `RuntimeException` and `SQLException`. These results indicate that *Monkey* has demonstrated the imperfection of manual testing and greatly enhances the objectivity of evaluating assignments.

In the typical testing process (companies), a few, mostly tens of applications are tested. However, in our case, we will test hundreds of the same applications several times a day, which can be time-consuming to perform. That's why we've run tests on the sample *Makacs* app to find out how long it takes to perform all the tests.

We've run tests on 3 real devices and 3 virtual devices (real device copies) where we tested sample app for 71 created tests. On virtual devices, we used Intel x86 Android system images, because of host's processor had mentioned instruction set. During the 10 measurements (Table II), we found that testing on a real device is approximately 2-3 times faster. Therefore, in addition to virtual devices (to perform tests on multiple devices), we will use the real device in particular. With 150 submitted solutions (expected number of students in 2017), our system can check on real devices on average for 6 hours 27 min 30 sec. It's quite long time, so we'll run tests probably every 12 hours.

TABLE II
Execution speed of 71 tests on different devices in 10 measurements.

Device	DT	Best Time	Avg. Time	Worst Time
Nexus 5, API 25, Android 7.1.1	VD	5 min 53 s	6 min 32 s	7 min 2 s
	RD	1 min 58 s	2 min 30 s	2 min 54 s
Prestigio 5453 DUO, API 19, Android 4.4	VD	2 min 12 s	2 min 52 s	4 min 1 s
	RD	2 min 23 s	2 min 59 s	3 min 30 s
Xiaomi Redmi 3s, API 23, Android 6.0	VD	4 min 12 s	5 min 48 s	6 min 8 s
	RD	1 min 52 s	2 min 16 s	2 min 36 s

DT - device type, VD - virtual device, RD - real device

During these tests, we encountered a problem with an unexpected throwing of exception that there is no activity opened. At the time of designing and testing the testing environment, *Espresso* developers were unable to resolve this problem, a few weeks later they fixed the bug. However, this issue still persisted on some devices. We explored that the problem occours while animations on a device are enabled, so it is necessary to turn off all device animations before running tests.

Other tests could not be tested on student solutions from previous years because the structure of the projects (classes, methods, IDs etc.) was not adapted to the proposed test environment.

VII. Future work

Improving the test process is an endless loop that can not be stopped. Based on our partial results, we will monitor students' responses to the test platform and customize test cases to bring maximum benefit to the student and teacher. The results from the real use in the course will be presented in the next paper where we expect feedback from approximately 150 students.

Given the problems with UI test speeds and their occasional unreliability with animations turned on (*Espresso*), we would like to compare and research different UI test frameworks in the future. Moreover, it might be interesting to require test cases by students in their assignments, than we could check these tests and use them to test other student assignments. Students will help us build tests and reveal deficiencies in testing process. About the evaluation of the tests made by the students is written by Smith et al. in paper [31] from 2017.

VIII. Conclusion

This experimental investigation has analyzed the most common mistakes of students who have become involved in previous runs of *Application Development for Smart Devices* course. Based on these mistakes and with the inspiration of the *test pyramid* concept, which seemed appropriate for use in the given course, we designed a test environment and used the appropriate test tools for the proposed tiers. In the end, we have found that the number of tests in our platform is not identical to the pyramid's ideology.

Of all the suggested tools, it was possible to perform only *Monkey* stress tests on solutions from past runs of the course. The results of the stress testing of a small sample proved to be appropriate and pointed to the inaccuracy of manual evaluation of the assignments. At the same time, we looked at the length of testing on real and virtual devices, and it was definitely more reliable and faster to perform tests on a real device. The authors also evaluated the advantages, disadvantages and experience with testing tools.

ACKNOWLEDGMENT

The authors would like to thank all the participants in the research, especially students and teachers of *Application Development for Smart Devices* course. At the same time we thank *Wirecard* company for sharing their research results, experiences with mobile testing and any other methodical help. This work was supported by project KEGA 047TUKE-4/2016 Integrating software processes into the teaching of programming.

REFERENCES

[1] M. Meeker, "Internet trends 2017," CODE CONFERENCE, Tech. Rep., 05 2017.

[2] StatCounter, "Operating system market share worldwide," 09 2017. [Online]. Available: http://gs.statcounter.com/os-market-share#monthly-201101-201708

[3] J. Esposito *et al.*, *Mobile Application Development*, C. Candelmo *et al.*, Eds. DZone, 2016, vol. 3.

[4] M. Cook, "The 50 most popular moocs of all time," 04 2015. [Online]. Available: http://www.onlinecoursereport.com/the-50-most-popular-moocs-of-all-time/

[5] Coursera Inc., "Courses and specializations," 2017. [Online]. Available: https://www.coursera.org/courses?_facet_changed_=true&domains=computer-science&languages=en&query=android

[6] edX Inc., "Android courses search," 2017. [Online]. Available: https://www.edx.org/course?search_query=android

[7] M. Madeja, "Innovative approaches in introductory programming courses," Master's thesis, Technical university of Košice, 05 2015.

[8] C. Domin, H. Pohl, and M. Krause, "Improving plagiarism detection in coding assignments by dynamic removal of common ground," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '16. New York, NY, USA: ACM, 2016, pp. 1173–1179. [Online]. Available: http://doi.acm.org/10.1145/2851581.2892512

[9] S. Mann and Z. Frew, "Similarity and originality in code: Plagiarism and normal variation in student assignments," in *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, ser. ACE '06. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 143–150. [Online]. Available: http://dl.acm.org/citation.cfm?id=1151869.1151888

[10] H. Munir, K. Wnuk, K. Petersen, and M. Moayyed, "An experimental evaluation of test driven development vs. test-last development with industry professionals," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '14. New York, NY, USA: ACM, 2014, pp. 50:1–50:10. [Online]. Available: http://doi.acm.org/10.1145/2601248.2601267

[11] L. V. Soham Mondal, "Android app development," 2017. [Online]. Available: https://www.springboard.com/learning-paths/android/

[12] D. C. Schmidt *et al.*, "Launch your android app development career," 2017. [Online]. Available: https://www.coursera.org/specializations/android-app-development

[13] F. A. Adrián Catalán, Noe Branagan, "Professional android app development," 2017. [Online]. Available: https://www.edx.org/course/professional-android-app-development-galileox-caad003x

[14] R. Pecinovský, "Zadávání a vyhodnocování úkolů při výuce oop," in *Počítač ve škole 2007*, Amaio Technologies, Inc. Nové Město na Moravě, Czech rep.: Počítač ve škole, 2007, pp. 1–4.

[15] J. Sugrue *et al.*, *Automated Testing*. DZone, 2017, vol. 1.

[16] Souce Labs, "Mobile app testing: Main challenges, different approaches, one solution," DZone, 09 2017.

[17] C. Q. Adamsen, G. Mezzetti, and A. Møller, "Systematic execution of android test suites in adverse conditions," in *Proceedings of the 2015 International Symposium on Software Testing and Analysis*, ser. ISSTA 2015. New York, NY, USA: ACM, 2015, pp. 83–93. [Online]. Available: http://doi.acm.org/10.1145/2771783.2771786

[18] B. Amen, S. Mahmood, and J. Lu, "Mobile application testing matrix and challenges," in *Computer Science & Information Technology*, vol. 5, 04 2015.

[19] M. Akourm, B. Falah, A. A. Al-Zyoud, S. Bouriat, and K. Alemerien, "Mobile software testing: Thoughts, strategies, challenges, and experimental study," in *(IJACSA) International Journal of Advanced Computer Science and Applications(ijacsa)*, vol. 7. SAI, 02 2016. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=7&Issue=6&Code=ijacsa&SerialNo=2

[20] Y. Wang and Y. Alshboul. (2016, 02) Mobile security testing approaches and challenges. Gainesville, Florida, USA. [Online]. Available: https://www.researchgate.net/publication/277132880_Mobile_Security_Testing_Approaches_and_Challenges

[21] H. Muccini, A. D. Francesco, and P. Esposito, "Software testing of mobile applications: Challenges and future research directions," in *2012 7th International Workshop on Automation of Software Test (AST)*, June 2012, pp. 29–35.

[22] C. Wilcox, "Testing strategies for the automated grading of student programs," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16. New York, NY, USA: ACM, 2016, pp. 437–442. [Online]. Available: http://doi.acm.org/10.1145/2839509.2844616

[23] T. Rajala, E. Kaila, R. Lindén, E. Kurvinen, E. Lokkila, M.-J. Laakso, and T. Salakoski, "Automatically assessed electronic exams in programming courses," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '16. New York, NY, USA: ACM, 2016, pp. 11:1–11:8. [Online]. Available: http://doi.acm.org/10.1145/2843043.2843062

[24] Y. Akahane, H. Kitaya, and U. Inoue, "Design and evaluation of automated scoring: Java programming assignments," *Int. J. Softw. Innov.*, vol. 3, no. 4, pp. 18–32, Oct. 2015. [Online]. Available: http://dx.doi.org/10.4018/IJSI.2015100102

[25] S. Gupta and S. K. Dubey, "Automatic assessment of programming assignment," in *ITCS, SIP, JSE-2012*, e. a. Natarajan Meghanathan, Ed. CS & IT, 2012, pp. 315–323.

[26] guru99.com. (2017, 08) Complete guide to android testing & automation. [Online]. Available: https://www.guru99.com/why-android-testing.html

[27] A. I. Wasserman, "Software engineering issues for mobile application development," in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, ser. FoSER '10. New York, NY, USA: ACM, 2010, pp. 397–400. [Online]. Available: http://doi.acm.org/10.1145/1882362.1882443

[28] C. Greb. (2016, 12) The 3 tiers of the android test pyramid. [Online]. Available: https://medium.com/android-testing-daily/the-3-tiers-of-the-android-test-pyramid-c1211b359acd

[29] Google Inc., *Test Your App*, 2017. [Online]. Available: https://developer.android.com/studio/test/index.html

[30] M. Madeja, "Testing of applications for os android," Master's thesis, Technical university of Košice, 04 2017.

[31] R. Smith, T. Tang, J. Warren, and S. Rixner, "An automated system for interactively learning software testing," in *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '17. New York, NY, USA: ACM, 2017, pp. 98–103. [Online]. Available: http://doi.acm.org/10.1145/3059009.3059022

Downsizing of Web Server Design Using Raspberry PI 3 Single Board Computer Platform

Branislav Madoš, Ján Hurtuk
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 040 01 Košice, Slovakia
{branislav.mados, jan.hurtuk}@tuke.sk

Eva Chovancová, Peter Feciľák, Dávid Bajkó
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 040 01 Košice, Slovakia
{eva.chovancova, peter.fecilak, david.bajko}@tuke.sk

Abstract—**Information and communication infrastructure (ICT) systems, partially or fully dedicated to the operation of internet, on the side of internet service providers and the side of internet users as well as on the communication infrastructure, form a significant share of the total ICT infrastructure on the global scale. Consumption of energy is an important aspect of the operation of this infrastructure not only in absolute numbers but also in comparison with office and home electrical appliances in general. That is why the reduction of its energy consumption draws attention. Contribution of the work described in this paper is the design of innovative web server platform providing Web Content Management System (WCMS) services using Raspberry Pi Single Board Computer (SBC) platform. Solution brings positive effects not only from ecological but also economical and computer security perspective. It is possible to see described design as the certain form of the web server downsizing in terms of size, energy consumption, associated heat dissipation and the price of the hardware platform and also downsizing of the web server software and data storage requirements with preservation of positive user experience. There is brief design description of the webserver presented in the paper, followed by results of selected performance and energy consumption tests.**

Keywords—server; web content management system; WCMS; single board computer; SBC; Raspberry Pi

I. Introduction

Information and Communication Technologies (ICT) are significant part of our professional and personal life. ICT helps create more effective and automated processes in industry, government, education and other fields. In personal life it can help build more effective environment for fulfilling of day-to-day duties and enhance communication and entertainment solutions. Spending on ICT reached USD 3.4 billion in 2016 with estimated 2.9% growth in 2017, when it will reach USD 3.5 billion according to the Gartner research. Within this amount we can find spending on data center systems with estimated USD 177 mld. and communication services representing another USD 1 410 mld. in the global scale [1]. Hardware devices and software are not the only items that must be paid. Total cost of ownership (TCO) includes also spending on electrical energy that is consumed by the ICT infrastructure and it is needed to evaluate also ecological aspects and carbon footprint from the society point of view.

Mid-range estimates are saying that ICT ecosystem consumes about 1500 TWh of electricity per year. Corresponding amount of energy is produced in Germany and Japan combined. The same amount of energy was used for lighting the whole planet in 1985 [2]. According to the International Energy Outlook 2016 (IEO2016), published by U.S. Energy Information Administration (EIA), world net electricity generation have been 21 600 TWh in 2012 [3]. Worldwide ICT electrical energy consumption reached 7% of the worldwide electrical energy production and internet itself was responsible for 2% of energy consumption worldwide.

There are some surprising comparisons with other electrical appliances that are documenting how energetically demanding is running some ICT equipment. For example medium size refrigerator consumes approximately 322 kWh of energy per year. Average iPhone consumes 361 kWh per year in comparison [2] and its consumption depends on factors like the use of wireless data transfer and the use of other wireless connections. There is statement in [4] that if the cloud were the country, it would have fifth largest electricity demand in the world, as you can see in the Fig. 1.

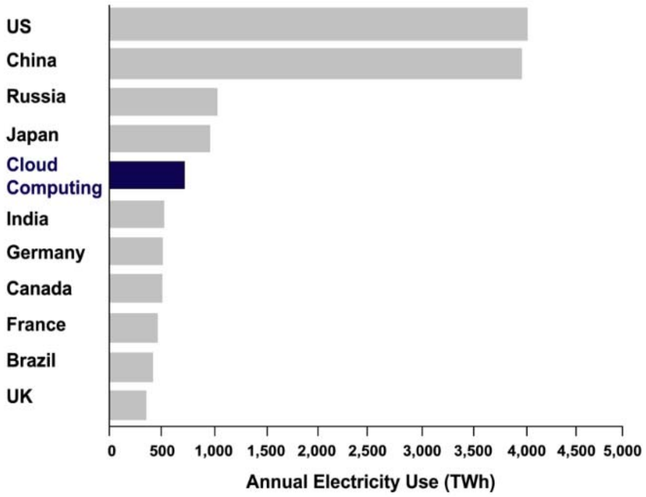


Figure 1 Consumption of electrical energy by different countries in comparison with the global cloud electric consumption. Source: Greenpeace International, How Clean is Your Cloud, April 2012

Web services are conceived as an essential component for promoting interoperability of business processes [5]. Data centres are the largest and most readily identifiable parts of the information and communication infrastructure. Energy consumption of data centres contributes to the world energy consumption by 1% to 2%. Centralisation of data storage and processing to data centres enables sharing of sources and allows replacement of less effective small local data centres especially in the segment of Small and Medium Enterprises (SME). Use of large centralised cloud solutions necessitates the increase of data transfer volumes on the other hand. Masanet and his colleagues in Lawrence Berkeley National Lab and the Northwestern University made publicly available Cloud Energy and Emissions Research Model (CLEER) [6] to assess the energy efficiency of the cloud in comparison with the existing digital systems. Most big data centres could slash their greenhouse gas emissions by 88 percent by switching to efficient, off-the-shelf equipment and improving energy management, according to the research [7].

Although data centres are great target of the effort to reduce energy consumption, there are other parts of ICT which are more hidden but in general represent great amount of electricity consumption and are bigger threat to the sustainability of cloud services. Nowadays we are transiting to the mobile internet and wireless networks consume far more energy than datacentres. Estimates are made that wireless networks consume 90% of energy in comparison of 9% of datacentres in mobile cloud. For example Centre for Energy-Efficient Telecommunications (CEET) is estimating, that wireless transport of 1GB of data consumes 2 kWh of electrical energy [8].

On the other hand, ICT helps to make many processes more effective or automated and carbon footprint of ICT infrastructure can be considered with broader effects in mind. Koomey gives an example, when infrastructure that is monitoring parking house with sensors which are connected to the internet is lowering the carbon footprint because it gives opportunity to car drivers to find parking place with assistance of the system without necessity of repeated circulation around the parking lot. 40 000 parking sposts in Los Angeles can be monitored using of only 15W of energy consumption [9].

When we think about making ICT more efficient, we can identify three main key areas:

- Data centres
- Access networks
- End user equipment

In our research we focused on the server part of the cloud infrastructure and proposed server platform with the use of the Raspberry Pi 3 model b single board computer as the hardware solution of the server with Linux operating system and in this research proposed server software. The strategy was in minimising hardware and software with the aim to preserve functionality of web server in the level that is suitable for websites for Small and Medium Enterprises (SME) but with less energy consumption, smaller form factor of the server hardware and with higher level of security in comparison with traditional rack servers. Modified version of Web Content Management Systems (WCMS) was also designed.

The structure of the paper is as follows.

Section 2 of the paper introduces proposed architecture of the web server using Raspberry Pi, Linux operating system and in this research developed minimized web server software solution along with the specialized Web Content Management System. Section 3 of the paper summarizes results of tests of the designed web server platform from different points of view comprising performance and energy consumption tests. Section 4, as the last part of the paper, represents conclusions resulting from tests summarized in the section 3 and outlines future work in the field.

II. Proposed architecture

Research is based on assumptions made on the basis of an examination of the current situation in the field of webhosting services and in the field of Web Content Management Systems. WCMS software is very popular and more than the half of websites is built with the use of this kind of software. Although there are hundreds of commercially available and open source WCMS software solutions, there are three of them – WordPress, Joomla and Drupal - which are very popular and represent more than half of WCMS installations. Popularity, robustness and broad field of hundreds free available extensions of different kinds attract the attention of not only ordinary users but also attackers, who decided to abuse vulnerabilities of web content management systems and misconducts of users in the phase of installation of this software. Many users who installed WCMS in their webhosting accounts became victims of the attackers. Webhosting providers reacted with option of the use of preinstalled secured custom versions of popular WCMS and are periodically scanning installations for malware [10]. Another very popular solution is the use of WCMS as the web service available after registration through the internet. VIX or Webnode can be an example of such commercially available services.

WCMS are often very space demanding, when it is necessary to allocate dozens of MB of disc space for their installation, and often are very computationally demanding especially when they compose webpage from information stored in their database for the first time.

Many customers are advantageously using WCMS as the tool which allows them to create websites without any knowledge of html or other necessary technology but on the other hand they do not use any benefits of dynamic websites. On the contrary users create very simple websites which comprise small number or subpages and can be very easily created as the static websites and stored as the source files on the disc instead of database.

Based on these assumptions we decided to build solution which divides backend and frontend of the WCMS into two different server platforms. Backend of the WCMS is programmed in PHP language and is using classical 1U Rack server with Linux operating system installed. Data is stored in MySQL database on SSD as it can be seen on Fig. 2. WCMS backend is available as the service with web interface to the relatively small number of users who have permission to modify the content of their respective websites.

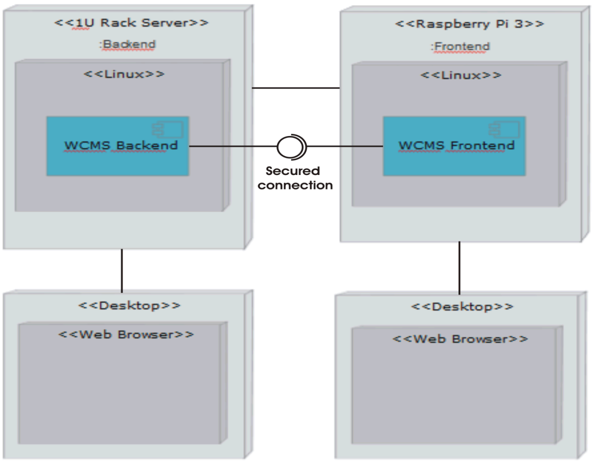


Figure 2 Deployment diagram of the designed architecture where WCMS backend is deployed with the use of classical 1U Rack Server and frontend deployed as the specialized server software and static versions of websites

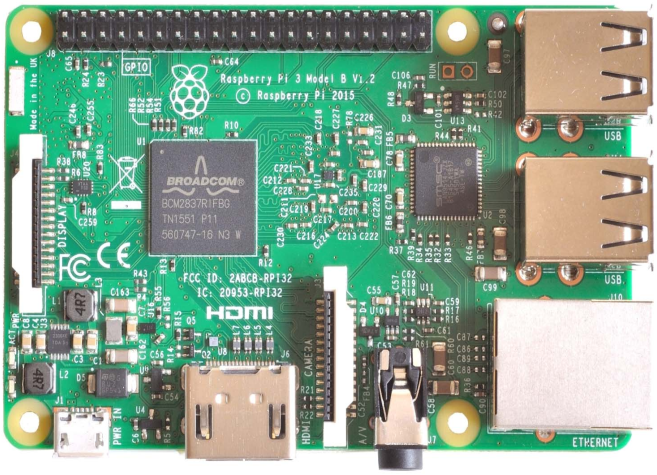


Figure 3 Raspberry Pi model 3 used as the frontend server designed as the single board computer (SBC) that represents complete computer architecture built on the single circuit board with the size of the credit card

This makes it easier to manage security issues in comparison with multiple customer installations of WCMS which make frontend and backend available through the same communication channels. Users are allowed to create websites as they are creating them in conventional WCMS and data is stored in conventional way in the database. Service allows users to create websites which are limited in the way that makes possible to convert them to the static form.

Backend and frontend of the WCMS are connected through the secured encrypted communication channel through which new and modified subpages of websites are sent from backend to the frontend of WCMS. This is the only communication channel through which requirements for modifications of websites can come to the frontend.

Frontend of the WCMS is realized with the use of Raspberry Pi 3 model b as the hardware platform. Linux as the operating system is installed along with the custom web server software designed as the part of this research. Websites are stored in their respective subdirectories in the static form on microSD card. Custom webserver software of the frontend is downsized to the minimal form which is necessary to allow processing of HTTP requests to deliver static webpages to user agents represented by web browsers and others, but it cannot receive any content from clients. No web forms or server side scripts are allowed and server is unable to send e-mails. These measures are intended to maximize security of this solution and to minimize demand for computational performance of the server. There is asymmetry in which each document that is forming particular website is read much more times that it is modified and therefore it is possible to advantageously use microSD card as the secondary storage for websites. It consumes small amount of energy in comparison with other secondary storage technology and has small form factor. On the other hand it disposes with relatively large space for storing websites data.

Advantage of this solution is that classical server technology, represented by energy demanding 1U rack server which has relatively big form factor and requires active cooling, is used only for backend of WCMS which plays role only in the early phase of website lifecycle when website is created and in relatively short times of website modifications. Small form factor Raspberry Pi server with small energy consumption and without need of active or extensive passive cooling is used in the phase of website lifecycle when it is accessible by website visitors. This part of website lifecycle is much longer and content is accessed much more times than in the creation/modification part of websites lifecycle.

III. Results

All tests were provided with the use of Raspberry Pi 3 model b that can be seen on Fig. 3 and microSD card used as the secondary storage, and 1U Intel rack server with two cores Intel Celeron processor, 2GB DDR2 ECC UDIMM operating memory, 2x 1000 Mbit/s Ethernet connection, 2x active cooling fans, power source 350W. Standard Linux operating system have been installed on both computers. Standard Apache server was used in case of 1U Intel Rack server and custom server software designed in this research have been used in the case of Raspberry Pi server.

Response time of the web server was measured for different amount of concurrent http requests in the range from 10 to 5000. Requests have been generated with the use of Webserver Stress Tool developed by PAESSLER. One measured value has been Time To First Byte (TTFB), average time for request and time for finish all requests. Each measurement was conducted 10 times. Average times for all measurements can be seen in Fig. 4 for Raspberry Pi and Fig. 5 for 1U Intel Rack server. Significant differences can be seen when 2000 and 5000 requests have been sent concurrently.

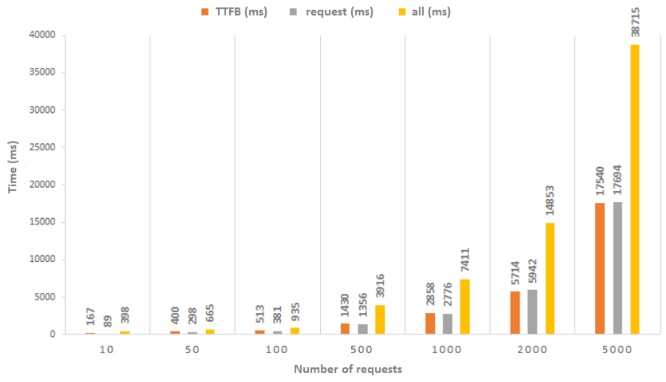


Figure 4 Average times for different amount of concurrent requests for Raspberry Pi 3 server.

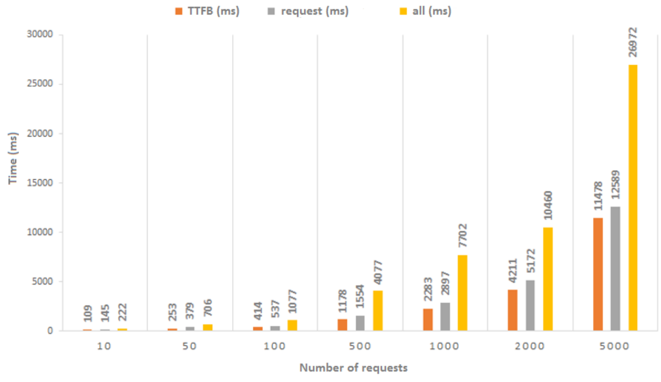


Figure 5 Average times for different amount of concurrent requests for 1U Intel Rack server.

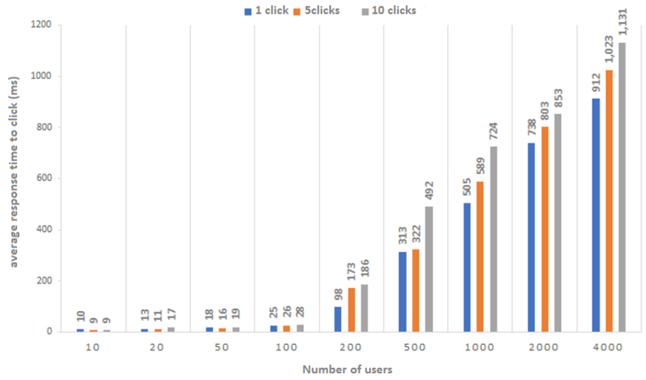


Figure 6 Average response time dependence on number of users and clicks for Raspberry Pi 3 model b.

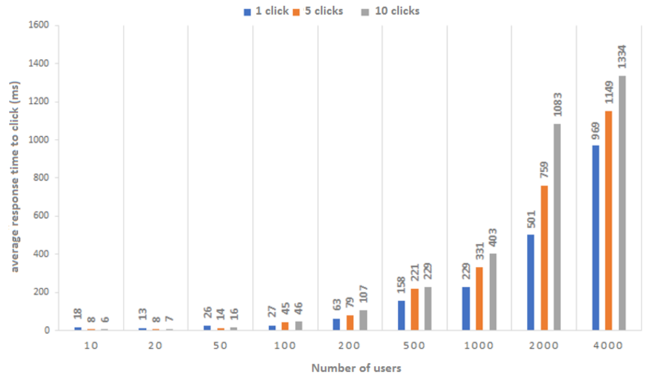


Figure 7 Average response time dependence on number of users and clicks for 1U Intel Rack server.

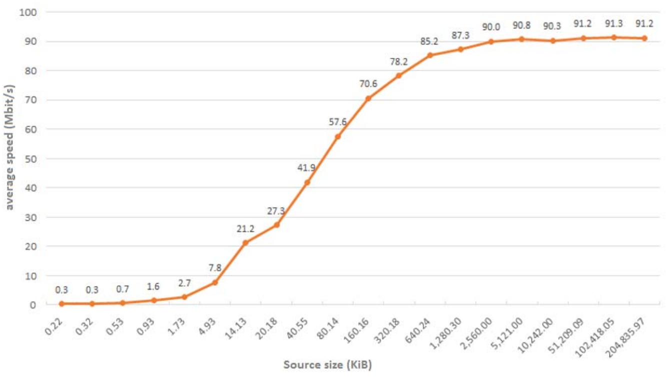


Figure 8 Data transfer speed according to the size of the file that has been transferred for Raspberry Pi 3 model b.

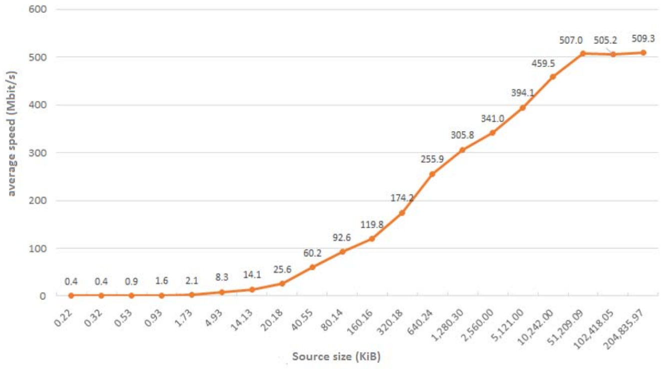


Figure 9 Data transfer speed according to the size of the file that has been transferred for 1U Intel Rack sever.

Webserver Stress Tool allows simulation of defined number of user clicks in defined time intervals and number of concurrent users that are sending requests. There are average response times in milliseconds according to number of clicks and number on concurrent users in the Fig. 6 for Raspberry Pi 3 and in the Fig. 7 for 1U Intel Rack server. Both graph representations have the same pattern, at the higher load Raspberry Pi has better results.

Another measurement was conducted when one source file was transferred from server to user agent and measurements were realized with different sizes of files as it is depicted on Fig. 8 for Raspberry Pi 3 and on Fig. 9 for 1U Intel Rack server. Raspberry Pi was able to send data with the highest speed of 91 Mbit/s. 1U Intel Rack server disposes with gigabit Ethernet connection and that is why the maximal speed of data transfer was approximately 510 Mbit/s.

Temperature measurement was performed with the use of Psensor utility. It is possible to see from Fig. 10 that Raspberry Pi 3 has higher temperature in all cases and maximum temperature difference has been 20 ° C. Raspberry Pi 3 has no active nor passive cooling subsystem, 1U rack server comprises 2 active coolers. It has been 10 000 concurrent http requests sent by the software Webserver Stress Tool. In the rest there were no http requests sent. Raspberry Pi processor load was 23% in average with 27% maximum and 1U Intel rack server processor load was 87% in average with 95% maximum.

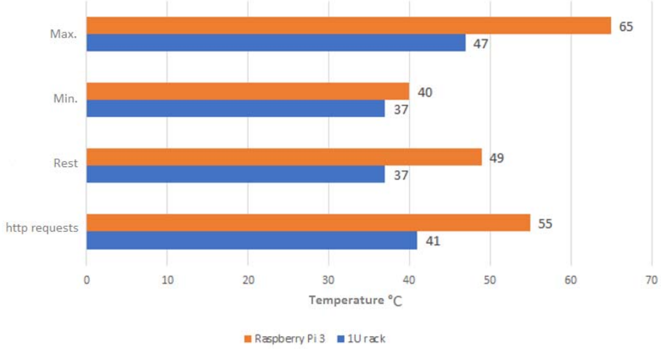


Figure 10 Temperature of Raspberry Pi 3 and 1U rack server performing the same tasks.

Energy consumption was measured by USB ampermeter in case of Raspberry Pi 3 and different measuring device Energy Meter was used in the case of 1U Intel rack sever. Energy consumption of Raspberry Pi in time when there were no http requests has been 1,25W and average energy consumption has been 2.5 W. Minimal energy consumption of 1U Intel server was 65W and maximal energy consumption has been 83 W.

IV. Conclusion

The paper introduced estimates on ICT infrastructure global energy consumption and discussed parts of ICT which have significant role in carbon footprint of ICT in the first part.

In the next section the paper outlined outcomes of our effort to downsize the web server design with the use of single board computer, particularly Raspberry Pi 3 model b. Design divides backend and frontend of the web content management system to two different server platforms, when frontend is downsized not only from the hardware but also from the software point of view. Backend of the WCMS is designed with the use of classical server hardware and software solutions. Advantages of the solution are in small form factor of frontend server with small energy consumption and better possibility to solve security issues. Drawback of the solution is in restrictions that are applied to websites, which are static and cannot use forms, server side scripts and cannot manage e-mails delivery.

The future work will be oriented to the security issues of proposed architecture and to the fault tolerance of the system which comprises many Raspberry Pi servers providing services concurrently.

Acknowledgment

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10. Project has been solved at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice.

References

[1] Gartner Says Global IT spending to Reach $3,5 trillion in 2017, Gartner Inc., Orlando, Florida, October 19, 2016, online: http://www.gartner.com/newsroom/id/3482917.

[2] P. Mills, The Cloud Begins With Coal, Big Data, Big Networks, Big Infrastructure and Big Power, An Overview of the Electricity Used by the Global Digital Ecosystem, Digital Power Group, August 2013.

[3] International Energy Outlook 2016 (IOE2016) With Projections to 2040, U.S. Energy Information administration (EIA), May 2016, Washington, online: https://www.eia.gov/outlooks/ieo/pdf/0484(2016).pdf.

[4] How Clean Is Your Cloud, Greenpeace, April 2012, online: http://www.greenpeace.org/international/en/publications/Campaign-reports/Climate-Reports/How-Clean-is-Your-Cloud/.

[5] O. Železník and Z. Havlice, MDA approach in embedded systems with strict real-time response and on-the-fly modeling requirements, - 2009. In: Acta Electrotechnica et Informatica. Vol. 9, No. 4 (2009), s. 30-36. - ISSN 1335-8243.

[6] Masanet et al., Cloud Energy and Emissions Research Model, Lawrence Berkeley National Lab and Northwestern University, online: http://cleermodel.lbl.gov/.

[7] E. Masanet, A. Shehabi and J. Koomey, Characteristics of low-carbon data centres, Nature Climate Change 3, 627–630 (2013), doi:10.1038/nclimate1786.

[8] The Power of Wireless Cloud, Centre for Energy-Efficient Telecommunications (CEET), online: http://www.ceet.unimelb.edu.au/publications/ceet-white-paper-wireless-cloud.pdf.

[9] J. Koomey, How green is the internet?, August 14, 2013, online: https://www.slideshare.net/jgkoomey/koomey-on.

[10] L. Vokorokos, A. Baláž and N. Ádám.: Events planning in intrusion detection systems, - 2007. In: Acta Electrotechnica et Informatica. Vol. 7, No. 4 (2007), s. 82-86. - ISSN 1335-8243.

High-level C++ Implementation of the Read-Copy-Update Pattern

Gábor Márton
Eötvös Loránd University,
Faculty of Informatics
Dept. of Programming Languages
and Compilers
H-1117 Pázmány Péter sétány 1/C
Budapest, Hungary
Email: martongabesz@gmail.com

Imre Szekeres
Budapest University of Technology
and Economics
Budapest, Hungary
Email: iszekeres.x@gmail.com

Zoltán Porkoláb
Eötvös Loránd University,
Faculty of Informatics
Dept. of Programming Languages
and Compilers
H-1117 Pázmány Péter sétány 1/C
Budapest, Hungary
Email: gsd@elte.hu

***Abstract*—Concurrent programming with classical mutex/lock techniques does not scale well when reads are way more frequent than writes. Such situation happens in operating system kernels among other performance critical multithreaded applications. Read copy update (RCU) is a well know technique for solving the problem. RCU guarantees minimal overhead for read operations and allows them to occur concurrently with write operations. RCU is a favourite concurrent pattern in low level, performance critical applications, like the Linux kernel. Currently there is no high-level abstraction for RCU for the C++ programming language. In this paper, we present our C++ RCU class library to support efficient concurrent programming for the read-copy-update pattern. The library has been carefully designed to optimise performance in a heavily multithreaded environment, in the same time providing high-level abstractions, like smart pointers and other C++11/14/17 features.**

I. Introduction

Read-copy-update is a concurrent design pattern [1], [2] which allows extremely low overhead for readers. Updates can happen concurrently with reads as they leave the old versions of the data structure intact; this way the already existing readers can finish their work. Thus, updates might require more overhead than reads and their effect might be delayed. In contrast to readers-writers lock [3] RCU does not block the writers if there are concurrent readers.

Classical RCU first appeared in the Linux kernel in 2002 [4], [5]. It provides the following reader side primitives: `rcu_read_lock()` and `rcu_read_unlock()`. Read-side critical sections may use `rcu_dereference()` to access RCU protected pointers.

On the update side we may use the `synchronize_rcu()` primitive and `rcu_assign_pointer()` to assign values to protected pointer. Pointers stored by `rcu_assign_pointer()` can be fetched from within read-side critical sections by `rcu_dereference()`.

The pseudo code in Figure 1 demonstrates how these primitives can be used to implement the lookup and the remove operations on a simple linked list of key-value pairs. This implementation is a simplified excerpt of McKenney's pre-BSD routing table example. With `rcu_read_lock()` and `rcu_read_unlock()` we indicate the reader side critical section. In this read side critical section we traverse through the list (`find()`) and once we found the key we return with the associated value. In the implementation of `find()` we have to use `rcu_dereference()` to access the elements in the list. It might happen that the key is not in the list, in that case we again close the critical section and then return with a special value indicating the element is not in the list.

```
SPINLOCK(lock);

Value lookup(List list, Key key) {
  Node* node;
  Value local_value;
  rcu_read_lock();
  // iterate over the list and return the value
  // of the found element
  if (node = find(list, key)) {
    local_value = node->value
    rcu_read_unlock();
    return local_value;
  }
  rcu_read_unlock();
  return not_found;
}

void remove (List list, Key key) {
  Node* node;
  spin_lock(lock);
  // iterate over the list and find the key
  if (node = find(list, key)) {
    remove_node(list, node);
    spin_unlock(lock);
    synchronize_rcu();
    free(node);
    return;
  }
  spin_unlock(lock);
}
```

Fig. 1. Usage of RCU in a linked list

In `remove()` we have to use a spin lock in order to protect the list from concurrent write operations. The block which is protected by the spin lock is the write-side critical section. We iterate over the list trying to find the key and if we found it then we unlink (`remove_node()`) it from the list. In the realization of the `remove_node()` we have to use the `rcu_assign_pointer()` primitive. After the removal, with the `synchronize_rcu()` primitive we wait all pre-existing RCU read-side critical sections to completely

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

finish. Then we can deallocate the list node which is no longer needed and this way can close the write-side critical section by releasing the lock.

Classic RCU requires that read-side critical sections obey the same rules obeyed by the critical sections of pure spinlocks: blocking or sleeping of any sort is strictly prohibited. Since 2002 many different RCU flavours have appeared in the Linux kernel which relax this strict requirement. Using realtime RCU [6]–[8] read-side critical sections may be preempted and may block while acquiring spinlocks. Sleepable RCU allows more, it permits arbitrary sleeping (or blocking) within RCU read-side critical sections [9], [10].

The different RCU flavours in the Linux kernel are naturally dependent on the kernel internals, for example on the scheduler. Obviously they cannot be used in user space. Userspace RCU (URCU) [11], [12] was created in 2009 and has a similar API to the kernel space RCU flavours. Userspace RCU has different variants and implementations. For instance the Quiescent-State-Based Reclamation RCU (QSBR) provides near-zero read-side overhead but the price of minimal overhead is that each thread in an application is required to periodically invoke `rcu_quiescent_state()` to announce that it resides in a quiescent state [13]. The general-purpose user space realization can be used in applications where we cannot guarantee that each threads will invoke `rcu_quiescent_state()` sufficiently often. However, this versatility has its own price, general-purpose RCU has to use memory barriers in the read-side. A third variant uses POSIX signals to eliminate these barriers, obviously this flavour cannot be used on non-POSIX systems.

URCU provides a low level C API, therefore it is more prone to errors in C++ programs than a well established high-level C++ API can be. For instance, it is easy to forget to call `rcu_read_unlock()` on all return paths. In URCU there is no automatic memory reclamation; to deallocate memory, first we have to use the `synchronize_rcu()` primitive.

In this paper we present an alternative implementation for user space RCU as a C++ smart pointer, thus there is no need to manually deallocate memory. Our realization provides a high-level abstraction C++ API to the users, so they can use a simple construct which is not prone to errors, still its performance is satisfying for most of the use cases. Our paper is organized as follows. In section II we present the steps which lead from using a mutex to the concept of a high-level smart pointer for the RCU semantics. We describe the details and difficulties with the implementation of the smart pointer in III. Section IV contains the description of our testing methods. We write about ongoing and future work in section V. Our paper concludes in VI.

II. Towards a Higher Level Abstraction for RCU

Let us suppose we have a collection that is shared among multiple readers and writers in a concurrent manner (Figure 2). It is a common way to make the collection thread safe by holding a lock until the iteration is finished (on the reader thread). This approach does not scale well, especially when reads are way more frequent than writes [5]. Instead of a simple `lock_guard` we could use a readers-writers lock [3], but that would scale badly as well, especially when we have multiple concurrent writers [5].

```
class X {
  std::vector<int> v;
  mutable std::mutex m;

public:
  int sum() const { // read operation
    std::lock_guard<std::mutex> lock{m};
    return std::accumulate(v.begin(), v.end(),
                           0);
  }
  void add(int i) { // write operation
    std::lock_guard<std::mutex> lock{m};
    v.push_back(i);
  }
};
```

Fig. 2. A shared collection

```
class X {
  std::shared_ptr<std::vector<int>> v;
  mutable std::mutex m;

public:
  X()
      : v(std::make_shared<
            std::vector<int>>()) {}
  int sum() const { // read operation
    std::shared_ptr<std::vector<int>>
        local_copy;
    {
      std::lock_guard<std::mutex> lock{m};
      local_copy = v;
    }
    // assume processing the data takes longer
    // than copying it
    return std::accumulate(local_copy->begin(),
                           local_copy->end(),
                           0);
  }
  void add(int i) { // write operation
    std::shared_ptr<std::vector<int>>
        local_copy;
    {
      std::lock_guard<std::mutex> lock{m};
      local_copy = v;
    }
    local_copy->push_back(i);
    {
      std::lock_guard<std::mutex> lock{m};
      v = local_copy;
    }
  }
};
```

Fig. 3. Using a shared pointer in the collection

The first idea to make it better is to have a shared pointer and hold the lock only until that is copied by the reader or updated by the writer (Figure 3). Now we have a race on the pointee itself during the write. So we need to have a deep copy (Figure 4). The copy construction of the underlying data (`vector<int>`) is thread safe, since the copy constructor parameter is a constant reference to `vector<int>`.

Still, there is one more problem: if there are two concurrent write operations then we might miss one of them. We should check whether the other writer had done an update after the actual writer has loaded the local copy. If it did then we should load the data again and try to do the update again. This leads to the idea of using an `atomic_compare_exchange` in a while loop. We could use an `atomic_shared_ptr` if

```
void add(int i) { // write operation
  std::shared_ptr<std::vector<int>> local_copy;
  {
    std::lock_guard<std::mutex> lock{m};
    local_copy = v;
  }
  auto local_deep_copy =
      std::make_shared<std::vector<int>>(
          *local_copy);
  local_deep_copy->push_back(i);
  {
    std::lock_guard<std::mutex> lock{m};
    v = local_deep_copy;
  }
}
```

Fig. 4. Deep copy

```
1  class X {
2    std::shared_ptr<std::vector<int>> v;
3
4  public:
5    X()
6        : v(std::make_shared<
7              std::vector<int>>()) {}
8    int sum() const { // read operation
9      auto local_copy = std::atomic_load(&v);
10     return std::accumulate(local_copy->begin(),
11                            local_copy->end(),
12                            0);
13   }
14   void add(int i) { // write operation
15     auto local_copy = std::atomic_load(&v);
16     auto exchange_result = false;
17     while (!exchange_result) {
18       // we need a deep copy
19       auto local_deep_copy =
20           std::make_shared<std::vector<int>>(
21               *local_copy);
22       local_deep_copy->push_back(i);
23       exchange_result =
24           std::atomic_compare_exchange_strong(
25               &v, &local_copy, local_deep_copy);
26     }
27   }
28 };
```

Fig. 5. Using atomic shared pointer

that was included in the current C++ standard, but until then we have to be satisfied with the free function overloads for shared_ptr (Figure 5). These free function overloads take a simple shared_ptr as a parameter and perform the specific atomic operations:

```
template <class T>
std::shared_ptr<T> atomic_load(
    const std::shared_ptr<T> *p);

template <class T>
bool atomic_compare_exchange_strong(
    std::shared_ptr<T> * p,
    std::shared_ptr<T> * expected,
    std::shared_ptr<T> desired);
```

Note, atomic_shared_ptr class template which would replace these free functions might be included in the C++20 standard [14]. Since both during the read operation and the write operation we do not modify the pointee the element type of the member shared_ptr can be changed to be a constant:

```
class X {
  std::shared_ptr<const std::vector<int>> v;
  // ...
};
```

In the write operation we do the update on the copy of the original pointee (line 22 of Figure 5) and not on the pointee of the member.

```
class X {
  rcu_ptr<std::vector<int>> v;

public:
  X()
      : v(std::make_shared<
            std::vector<int>>()) {}
  int sum() const { // read operation
    std::shared_ptr<const std::vector<int>>
        local_copy = v.read();
    return std::accumulate(local_copy->begin(),
                           local_copy->end(),
                           0);
  }
  void add(int i) { // write operation
    v.copy_update([i](std::vector<int> *copy) {
      copy->push_back(i);
    });
  }
};
```

Fig. 6. Usage of rcu_ptr

We might notice that we can move construct the third parameter of atomic_compare_exchange_strong, therefore we can spare a reference count increment and decrement:

```
exchange_result =
    std::atomic_compare_exchange_strong(
        &v, &local_copy,
        std::move(local_deep_copy));
```

Regarding the write operation, since we are already in a while loop we could replace atomic_compare_exchange_strong with atomic_compare_exchange_weak. That can result in a performance gain on some platforms [15], [16]. However, atomic_compare_exchange_weak can fail spuriously[1]. Consequently, we might do the deep copy more often than needed if we used the weak counterpart.

In the current form of class X nothing stops an other programmer (e.g. a naive maintainer of the code years later) to add a new reader operation, like this:

```
int another_sum() const {
  return std::accumulate(v->begin(), v->end(),
                         0);
}
```

This is definitely a race condition and a problem. To avoid this user error and to hide the sensitive technical details we created a smart pointer which we named as rcu_ptr. This smart pointer provides a general higher level abstraction above atomic_shared_ptr. Figure 6 represents how can we use rcu_ptr in our running example. The read() method of rcu_ptr returns a shared_ptr<const T> by value, therefore it is thread safe. The existence of the shared_ptr in the scope enforces that the read object will live at least until this read operation finishes. By using the shared pointer this way, we are free from the ABA problem [17], [18] since the memory address associated with the object cannot be reused until the object itself is reclaimed [19]. The copy_update() method receives a lambda. This lambda is called whenever an update needs to be done, i.e. it will be called continuously until the update is successful. The lambda

[1] Spurious failure enables implementation of compare-and-exchange on a broader class of machines, e.g., load-locked store-conditional machines [15]

```
1  template <typename T> class rcu_ptr {
2    std::shared_ptr<const T> sp;
3
4  public:
5    rcu_ptr() = default;
6    ˜rcu_ptr() = default;
7
8    rcu_ptr(const rcu_ptr &rhs) = delete;
9    rcu_ptr &
10   operator=(const rcu_ptr &rhs) = delete;
11   rcu_ptr(rcu_ptr &&) = delete;
12   rcu_ptr &operator=(rcu_ptr &&) = delete;
13
14   rcu_ptr(const std::shared_ptr<const T> &sp_)
15       : sp(sp_) {}
16   rcu_ptr(std::shared_ptr<const T> &&sp_)
17       : sp(std::move(sp_)) {}
18
19   std::shared_ptr<const T> read() const {
20     return std::atomic_load_explicit(
21         &sp, std::memory_order_consume);
22   }
23
24   void
25   reset(const std::shared_ptr<const T> &r) {
26     std::atomic_store_explicit(
27         &sp, r, std::memory_order_release);
28   }
29   void reset(std::shared_ptr<const T> &&r) {
30     std::atomic_store_explicit(
31         &sp, std::move(r),
32         std::memory_order_release);
33   }
34
35   template <typename R>
36   void copy_update(R &&fun) {
37
38     std::shared_ptr<const T> sp_l =
39         std::atomic_load_explicit(
40             &sp, std::memory_order_consume);
41
42     std::shared_ptr<T> r;
43     do {
44       if (sp_l) {
45         // deep copy
46         r = std::make_shared<T>(*sp_l);
47       }
48
49       // update
50       std::forward<R>(fun)(r.get());
51
52     } while (
53       !std::
54         atomic_compare_exchange_strong_explicit(
55                 &sp, &sp_l,
56                 std::shared_ptr<const T>(
57                     std::move(r)),
58                 std::memory_order_release,
59                 std::memory_order_consume));
60   }
61 };
```

Fig. 7. The rcu_ptr class template

receives a `T*` for the copy of the actual data. We can modify the copy of the actual data inside the lambda.

III. Smart Pointer for RCU semantics

In Figure 7 we present the implementation of the `rcu_ptr` class template. We provide a default constructor and a default destructor (lines 5 and 6). The move and copy operations are deleted (lines 8-12) because `rcu_ptr` is essentially a wrapper around an atomic type (we plan to support `atomic_shared_ptr` as soon as it is included in the standard). And all atomic types are neither copyable nor movable (because there is no sense to assign meaning for an operation spanning two separately atomic objects) [20], [21].

We can create an `rcu_ptr` from an lvalue or rvalue reference of `shared_ptr<const T>` (lines 14-17). These functions just simply copy or move their parameter into the member `shared_ptr`. There is no need to make these constructors thread safe, because the construction can be done only by one thread.

Lines 24-33 is the realization of the `reset()` methods which receive a `shared_ptr<const T>` as an lvalue or rvalue reference parameter. We can use it to reset the wrapped data to a new value independent from the old value (e.g. `vector.clear()`). Actually, with the parameter we overwrite the currently contained `shared_ptr`. The overwrite has to be an atomic operation in order to protect the member from concurrent `reset()` calls.

In lines 19-22, the `read()` method atomically loads the member `shared_ptr` and returns with a copy of that. The `copy_update()` function template (lines 35-60) receives an rvalue reference to an instance of a callable type. First we create a local copy of the member as `sp_l` (lines 38-40). If this local copy is set (i.e the `rcu_ptr` instance is initialized) then we create a deep copy, that is we copy the pointee itself and we create a new `shared_ptr<T>` (denoted as `r`) pointing to the copy (lines 44-47). Note, that this is a non-constant shared pointer. On line 50 we call the callable and we pass a non-constant pointer to the new copy as a parameter. Then in lines 53-59 we exchange the member shared pointer with a `shared_ptr` to the deep copy if we find that the member still points to the same object of which we created the copy. If it turns out that is not the case (i.e. another thread was faster), then we repeat the whole deep copy update sequence until we succeed (line 43). The callers of the `copy_update()` function must be aware that in case of an unset (or default initialized) `rcu_ptr` the callable will be called with a null pointer as an argument. Also, a call expression with this function is invalid, if the wrapped data type (`T`) is a non-copyable type.

A. *Memory Ordering*

A `memory_order_release` store is said to *synchronize with* a `memory_order_acquire` load if that load returns the value stored or in some special cases, some later value [15], [22]. When a `memory_order_release` store synchronizes with a `memory_order_acquire` load, any memory reference preceding the `memory_order_release` store will *happen before* any memory reference following the `memory_order_acquire` load [15], [22]. This property allows a linked structure to be locklessly traversed by using `memory_order_release` stores when updating pointers to reference new data elements and by using `memory_order_acquire` loads when loading pointers while locklessly traversing the data structure [22]. A `memory_order_release` store is *dependency ordered before* a `memory_order_consume` load when that load returns the value stored, or in some special cases, some later value [15], [22]. Then, if the load carries a dependency to some later memory reference, any memory reference preceding the `memory_order_release` store will happen before that later memory reference [15], [22]. This means that when there is dependency ordering, `memory_order_consume` gives the same guarantees that memory_order_acquire does, but at lower cost [22].

In the classical RCU, the rcu_dereference() primitive implements the notion of a dependency ordered load, which suppresses aggressive code-motion compiler optimizations and generates a simple load on any system other than DEC Alpha, where it generates a load followed by a memory-barrier instruction. The rcu_assign_pointer() primitive implements the notion of store release, which on sequentially consistent and total-store-ordered systems compiles to a simple assignment [11].

In our implementation of rcu_ptr::copy_update() function we can also use the release and consume semantics. We cannot use relaxed ordering because in case of that if the fun is inlined and fun itself is not an ordering operation or it does not contain any fences then the load or the compare_exchange might be reordered into the middle of fun. Also we need to "see" the latest updates so we can copy and update the "most recent" version. Though, there is a data dependency chain: sp_l->r->compare_exchange(...,r). So if all the architectures were preserving data dependency ordering, than we would be fine with relaxed. However, some architectures do not preserve data dependency ordering (e.g. DEC Alpha), therefore we need to explicitly state that we rely on that neither the CPU nor the compiler will reorder data dependent operations. This is what we express with the consume-release semantics. Consequently, during all the atomic load operations in the rcu_ptr class template we can use memory_order_consume and during all atomic store operations (including the read-modify-write operation) we use memory_order_release. If the definition of the fun callable is unseen by the compiler (i.e. it is defined in an other translation unit) then the user have to annotate the declaration of the callable with the [[carries_dependency]] attribute [15]. Otherwise, the compiler may assume that the dependency chain is broken during the call and consequently it would fall back to the safer but less efficient acquire semantics [15].

Unfortunately the consume memory order is temporarily deprecated in C++17. It is widely accepted that the current definition of memory_order_consume in the C++11/14 standard is not useful. All current compilers essentially map it to memory_order_acquire. The difficulties appear to stem both from the high implementation complexity and from the fact that the current definition uses a fairly general definition of "dependency" [22], [23]. As such, the consume ordering has to be redefined. While this work is in progress, hopefully ready for the next revision of C++, users are encouraged to not use this ordering and instead use acquire ordering, so as to not be exposed to a breaking change in the future. As for our rcu_ptr, in order to reach the consume semantics we plan to use hardware specific instructions in the future to overcome the mentioned problem.

B. Lock Free *atomic_shared_ptr*

Our rcu_ptr relies on the free functions overloads with the atomic_ prefix [15, section 20.8.2.6] for std::shared_ptr. It would be nice to use an atomic_shared_ptr [14], but currently that is still in experimental phase. We use atomic shared_ptr operations which are implemented in terms of a spinlock (that is how it is implemented in the currently available standard libraries). Having a lock-free atomic_shared_ptr would be really beneficial. However, implementing a lock-free atomic_shared_ptr in a portable way can have extreme difficulties [24]. Though, it might be easier on architectures where the double word CAS operation is available as a CPU instruction as we can see that with Anthony Williams implementation [25].

IV. Correctness and Testing

To validate the correctness of our data structure we used different testing methods. We executed unit tests in a sequential manner (i.e. no parallel execution) to validate the basic behaviour of the class template. We used oriented stress testing [26] and sanitizers from the LLVM/Clang infrastructure [27] to verify behaviour during concurrent execution. During our stress tests we focused on pairs of public methods of rcu_ptr and we executed these functions from different threads. We executed the operations in a loop on each thread and we added random delays in between each calls. This way we tested different execution timings and we could make race windows slightly larger.

V. Future Work

It is our ongoing work to create concrete and precise performance measurements. We aim to measure the performance of our rcu_ptr on a weekly ordered architecture like ARMv7. Our target is to do measurements on different dimensions because the performance may depend on the architecture, the number of reader or writer threads, the ratio of the readers/writers, the size of the wrapped data, etc. Also, we plan to compare our implementation with URCU and readers-writers lock in different use cases. The complexity and the huge variegation of possible measurements drive us to publish the future results in a different paper.

VI. Conclusion

RCU is a technique in concurrent programming which is getting used more and more often nowadays. It has been introduced in the Linux kernel first, but the efficiency of the technique became proven so people demanded an implementation which could be used in user space too. The current available user space RCU solutions do not provide a mechanism for automatic memory reclamation, also they provide a low level C API, which may be prone to errors. In this paper we presented a high-level C++ implementation for the read-copy-update pattern, which provides automatic memory deallocation while providing a safer and hard-to-misuse API.

Acknowledgment

The authors would like to thank to Péter Bolla for having valuable discussions about the implementation, and the public interface. We would like to thank also to for Máté Cserna, for his really helpful comments on the library implementation.

References

[1] P. E. McKenney and J. D. Slingwine, "Read-copy update: Using execution history to solve concurrency problems," in *Parallel and Distributed Computing and Systems*, 1998, pp. 509–518.

[2] P. E. McKenney, J. Appavoo, A. Kleen, O. Krieger, R. Russell, D. Sarma, and M. Soni, "Read-copy update," in *AUUG Conference Proceedings*. AUUG, Inc., 2001, p. 175.

[3] J. M. Mellor-Crummey and M. L. Scott, "Scalable reader-writer synchronization for shared-memory multiprocessors," *SIGPLAN Not.*, vol. 26, no. 7, pp. 106–113, Apr. 1991. [Online]. Available: http://doi.acm.org/10.1145/109626.109637

[4] P. E. McKenney and J. Walpole, "What is RCU, fundamentally?" December 2007, available: http://lwn.net/Articles/262464/ [Viewed December 27, 2007].

[5] P. E. McKenney, *Is Parallel Programming Hard, And, If So, What Can You Do About It?* Corvallis, OR, USA: kernel.org, 2010. [Online]. Available: http://kernel.org/pub/linux/kernel/people/paulmck/perfbook/perfbook.html

[6] P. McKenney, "The design of preemptible read-copy-update," October 2007, available: http://lwn.net/Articles/253651/ [Viewed October 25, 2007].

[7] P. E. McKenney, D. Sarma, I. Molnar, and S. Bhattacharya, "Extending rcu for realtime and embedded workloads," in *Ottawa Linux Symposium, pages v2*, 2006, pp. 123–138.

[8] P. E. McKenney and D. Sarma, "Adapting rcu for real-time operating system usage," Oct. 23 2007, uS Patent 7,287,135.

[9] P. E. McKenney, "Sleepable RCU," October 2006, available: http://lwn.net/Articles/202847/ Revised: http://www.rdrop.com/users/paulmck/RCU/srcu.2007.01.14a.pdf [Viewed August 21, 2006].

[10] D. Guniguntala, P. E. McKenney, J. Triplett, and J. Walpole, "The read-copy-update mechanism for supporting real-time applications on shared-memory multiprocessor systems with Linux," *IBM Systems Journal*, vol. 47, no. 2, pp. 221–236, May 2008.

[11] M. Desnoyers, P. E. McKenney, A. S. Stern, M. R. Dagenais, and J. Walpole, "User-level implementations of read-copy update," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 2, pp. 375–382, 2012.

[12] M. Desnoyers, "[RFC git tree] userspace RCU (urcu) for Linux," February 2009, http://lttng.org/urcu.

[13] T. E. Hart, P. E. McKenney, A. D. Brown, and J. Walpole, "Performance of memory reclamation for lockless synchronization," *J. Parallel Distrib. Comput.*, vol. 67, no. 12, pp. 1270–1285, 2007.

[14] H. Sutter, "Atomic smart pointers, rev. 1," ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++, Tech. Rep. n4162, Oct. 2014.

[15] ISO, *ISO/IEC 14882:2014 Information technology — Programming languages — C++*. Geneva, Switzerland: International Organization for Standardization, 2014.

[16] stackoverflow.com, "Understanding std::atomic::compare_exchange_weak() in c++11," 2017. [Online]. Available: https://goo.gl/jwjgGC

[17] R. K. Treiber, *Systems programming: Coping with parallelism*. International Business Machines Incorporated, Thomas J. Watson Research Center, 1986.

[18] D. Dechev, P. Pirkelbauer, and B. Stroustrup, "Understanding and effectively preventing the aba problem in descriptor-based lock-free designs," in *Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2010 13th IEEE International Symposium on*. IEEE, 2010, pp. 185–192.

[19] A. Williams, "Why do we need atomic_shared_ptr?" August 2015, available: https://www.justsoftwaresolutions.co.uk/threading/why-do-we-need-atomic_shared_ptr.html.

[20] Anthony Williams, *C++ concurrency in action: practical multithreading*. Manning Publ., 2012.

[21] stackoverflow.com, "Why are std::atomic objects not copyable?" 2017. [Online]. Available: https://goo.gl/fvuY3f

[22] P. E. McKenney, T. Riegel, J. Preshing, H. Boehm, C. Nelson, O. Giroux, and L. Crowl, "Towards implementation and use of memory_order_consume," ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++, Tech. Rep. P0098R0, 2015.

[23] H.-J. Boehm, "Temporarily deprecate memory_order_consume," ISO/IEC JTC 1, Information Technology, Subcommittee SC 22, Programming Language C++, Tech. Rep. P0371R0, May 2016.

[24] M. McCarty, "Implementing a lock-free atomic_shared_ptr," 2016, cppNow 2016. [Online]. Available: https://goo.gl/qErf1h

[25] A. Williams, "Implementation of a lock-free atomic_shared_ptr class template as described in n4162," 2016. [Online]. Available: https://bitbucket.org/anthonyw/atomic_shared_ptr

[26] M. Desnoyers, "Proving the correctness of nonblocking data structures," *Communications of the ACM*, vol. 56, no. 7, pp. 62–69, 2013.

[27] llvm.org. (2017) clang: a c language family frontend for llvm. [Online]. Available: http://clang.llvm.org

Measurement Effect of Visibility in Experimental FSO system

*Michal Márton, *Ľuboš Ovseník
*Department of Electronics and Multimedia Communications of University of Technology Košice,
michal.marton@tuke.sk, lubos.ovsenik@tuke.sk

**Michal Špes
**Department of Electrical Engineering of University of Technology Košice
michal.spes@tuke.sk

Abstract—**Effect of visibility attenuation in FSO (Free Space Optics) has dominant impact on quality of received signal. Therefore, it is important to monitor this parameter, which is crucial for management of switching between optical communication and radio communication. The point of this paper includes design of measurement and analysis of results of visibility measurement on TUKE (Technical University of Kosice), where is FSO system deployed. Measurement of different weather parameters is running for this system on TUKE. Optical and radio channels are switching on the fly depending on base obtained by measurement of visibility values. We will describe possible methods of measurement of one weather parameter. This key parameter is visibility. Different measurement approaches provide different processing speeds. Faster and more accurate switching ensures stable system without disruption.**

Keywords—FSO, measurement of visibility, reliability

I. Introduction

The prediction of weather changes is on high level. Meteorological devices which give actual results of weather prediction are widely used. It is very important to secure stable FSO communication link. In optical communications in free space two types of atmospheric interactions are well known. First type of atmospheric interaction occurs in case when optical beam spread interacting's with atmospheric aerosols and molecules. Second case occurs when optical turbulence is caused by fluctuations intensity laser beam. These nonlinear influences have dominant effect of quality of transmitted optical signal between optical FSO terminals, which create experimental FSO communication link. In case there are very bad conditions for usage of optical communication link has to switch to backup link because the value of attenuation is high. Size of atmospheric particles is very variable and each particle of other type of weather has different lifetime. The small atmospheric particles have very long lifetimes unlike high atmospheric particles which have very short lifetimes. Interaction of optical beam and atmospheric particles creates influence which is named as scattering. Known types of scattering are Mie and Rayleigh scattering. Prevention before the interaction of optical beams with atmospheric particles is necessary to minimizing of using FSO communication link to communication between nodes in cases when there are very bad weather conditions and the line of sight doesn´t exist. Switching between primary FSO and secondary backup link must be fast and adaptive. It is very important to monitor weather conditions for line of sight communication links. We monitor and evaluate weather conditions for FSO link on TUKE. The most important weather condition for stable FSO system is visibility. Different types of visibility measurement this parameter are the point of this article [1].

II. The Impact of Weather on FSO Transmission Quality

The prime types of weather which significantly affect quality of FSO system transmission are haze, fog and dense fog. These weather conditions have other dominant effects for used different wavelengths. Interactions of atmospheric particles with optical beam take effect by increase of attenuation which causes extinction of this beam. By extinction of optical beam occurs when weather conditions are very bad and attenuation on transmission path is significant. Transmittance is defined by relation (1).

$$T_v(s) = \frac{I_v(s)}{I_{v,0}} = e^{-\beta_v s} \tag{1}$$

In relation (1) β_v corresponds with total extinction coefficients of aerosol absorption, molecular absorption and molecular scattering [1].

TABLE I. Typical Values of Total Extinction Coefficient

Air condition	*Clear*	*Haze*	*Fog*	*Dense Fog*
$\beta_v s$ [km^{-1}]	0.1	1.0	>10	391

With using values of extinction coefficient from TABLE .I is possible to determine examined value of attenuation [1]. Channel loss in the transmission path is represented through attenuation in dB units. Attenuation was calculated for each type of air conditions from equations (2-5).

In case of very good weather conditions during transmission the value of attenuation L_a is equal to -0.22dB. By very good conditions we understand clear sky. Attenuation is calculated for distance between communication points of 500m.

$$L_a = 10\log_{10}(T) = 10\log_{10}(e^{-0.1*0.5}) = -0.22dB \tag{2}$$

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

For haze the value of attenuation is calculated by equation (3) for the same distance.

$$L_a = 10\log_{10}(T) = 10\log_{10}(e^{-1*0.5}) = -2.17dB \qquad (3)$$

The attenuations for fog and dense fog are calculated with usage of (4) and (5), respectively. The distance stays the same as in previous cases.

$$L_a = 10\log_{10}(T) = 10\log_{10}(e^{-10*0.5}) = -21.71dB \qquad (4)$$

$$L_a = 10\log_{10}(T) = 10\log_{10}(e^{-391*0.5}) = -849.05dB \qquad (5)$$

These results show sensitivity to atmospheric particles which interact with optical beam which is then attenuated and in several cases it is extinct. The most important is Mie scattering for FSO system because impact of this phenomena to SNR (Signal to Noise Ratio) does not depend of used wavelength of optical beam. Hence the creation of this effect cannot be avoided [1-3].

III. Experimental Scenario

Free space optic through wireless channel offers full duplex communication. Nowadays FSO systems are very popular because they enable connections of many points between which the connections based on physical optical fibers are not possible or they are more expensive. Distance between points which could communicate is in range from few meters to several kilometers. Communication with using wireless channel as optical transmission path offers many advantages which make FSO a very attractive system. Speed of transmission, using non-licensed transmission band, no need for expensive optical components and fibers are a few from the many advantages which this system offers. With raising trend of FSO development and it allows reaching bitrates higher than 10Gbps on distance of few kilometers without usage of physical optical fibers. The next advantage is installation which is very easy compared with standard optical communication system. FSO could be applied as backbone connection however it has to have backup line which would be used during very bad weather conditions when attenuation rises to critical values. Value of BER (Bit Error Rate) must be BER=10^{-9} or better and then the system can be considered as stable system with good error probability. FSO system is based on two optical terminals which allow receiving, transmission of optical signal through wireless channel. The optical terminals work on different wavelengths. Used wavelength is selected by type of used laser. In one area can be used more FSO systems working on same wavelength without errors. In several cases independent systems working on different wavelengths are chosen. The possible wavelengths are in optical transmission windows near 850nm, 1310nm and 1550nm. Each optical system is can work on one of these wavelengths [4].

Communication between two points is made through switch which makes decision about used transmission path. Decision making must be quick and it should take into account values gained from visibility sensors. In case of bad weather conditions the FSO link should be replaced by backup connection. Possible solutions for backup connections include one of newer WiFi standards-802.11ac with using MIMO system in closer areas. This standard offers speed up to 7Gbps in 5GHz band [5].

Function of electro-optical convertor and reverse optical-electro convertor is conversion of signal to form which is supported by device on the next point.

Type of used optical terminal installed on buildings is Flightstrata 155E. Terminal Flightstrata 155E contains four optical lasers working on 850nm wavelength. The number of used lasers increases reliability compared to optical system which uses one optical laser.

TABLE II. Parameters of FSO Terminal

Lightpointe Flightstrata 155E	
Parameters	**Values**
Wavelength	850nm
TX Power	160mW
RX Power (Sensitivity)	-30dBm
Diameter of receiving lens	8cm
Directivity	2mrad

TABLE II shows parameters of terminal used in experimental model of FSO system. Design of experimental FSO model installed on TUKE is shown on Fig.1 [6].

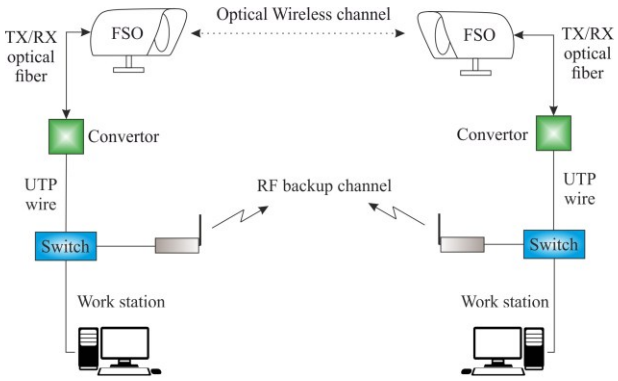


Fig. 1 Experimental model of FSO system.

FSO terminal Lightpointe Flightstrata 155E is shown on Fig.2.

Fig. 2 Flightstrata 155E [6].

IV. Measuring of Visibility

The need for ensuring high reliability is the main objective of measuring of visibility. Visibility is one of parameters for switching between FSO system and backup communication system. Speed of switching is very important, when the duration of switching is short then outage is not noticed by users. Nowadays many reliable possibilities of visibility measurement are available. Accurate methods of visibility measurement are offered by devices used in airports and in professional Meteorological stations placed near to airports. On TUKE we have own measuring device which is based on several sensors.

A. Transmissometer MITRAS

Measurement device MITRAS made in Finland in Vaisala is designed to measure RVR (Runway Visual Range). Transmissiometer MITRAS is professional equipment used in professional meteorological stations and airports. Device is based on light transmitter and light receiver, which are placed opposite each other. Examined parameter is attenuation of white light transmitted between two devices. Resulting values gained from measurement are converted to MOR (Meteorological Optical Range) values. This type of transmissiometer allows measuring MOR in range of 40-2000m. Shortest possible period between successive measurements is 16s. MITRAS can be configured for using on single baseline measurement or double baseline. Single baseline offers measurement of visibility on the same transmission path on which FSO operates Example using transmissiometer MITRAS in single baseline is displayed on Fig.3 [7].

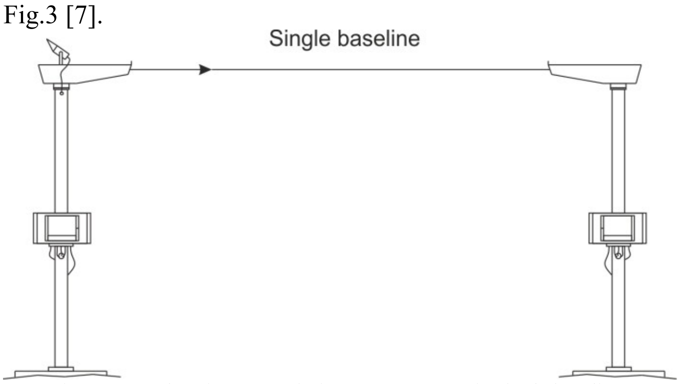


Fig. 3 Example using transmissiometer MITRAS in single baseline.

TABLE III. MEASUREMENT RANGES ACHIEVED BY SINGLE BASELINE [7]

Single baseline		
Baseline	**MOR range**	**RVR range**
35m	20-1500m	100-1500m
50m	40-2000m	150-2000m
75m	50-3000m	200-3000m
100m	70-5000m	300-5000m
200m	150-10 000m	500-10000m

In TABLE III shows achievable ranges for single baseline configuration. Trasmissiometer can be placed in different distances from range 10 to 200m. This range defines measure range for MOR and RVR. RVR depends on runway lighting if is used on airport runway.

B. Transmissiometer LT31

Transmissiometer LT31 can be considered as an upgraded version of transmissiometer MITRAS with a few differences. It is professional device for measuring visibility with high reliability. The MOR range is from 10 to 10 000m. This range covers full RVR range which is necessary for airports. The main difference is using single baseline configuration for achieving full measure range in contrast with transmissiometer MITRAS. Coverage of full measuring range with MITRAS needs application of double baseline configuration.

Transmissiometer LT31 is based on measuring of backscattered optical beam and using LED optical laser with white light. White light is the best for transmission measure [8]. This equipment can be adjusted by trained personnel. The other method is automatic alignment with integrated forward scatter sensor. This method was patented by Vaisala [8].

TABLE IV. MEASUREMENT RANGES OF LT31 [9]

Measurement range of LT31		
Baseline	**MOR range**	**Transmittance range**
35m	10-10 000m	<0.01-100%
50m	25-10 000m	<0.02-100%
75m	37.50-10 000m	<0.02-100%

TABLE IV shows values of achieved optical range for measuring visibility. Optical device LT31 is displayed on Fig. 4.

Fig. 4 Transmissiometer LT31.

C. Visibility measurement on TUKE

Professional transmissiometers as MITRAS and LT31 are very expensive and they are used only in professional meteorological stations (SHMU – Slovak Hydrometeorological Institute) and airports. We need to calculate visibility by another method. This method is monitoring fog and then calculation of visibility from achieved values. Bad values of visibility are in many cases caused by fog. Very important parameter which causes fog is LWC (Liquid Water Content). LWC in fog will reach values in wide range. Types of fog are represented by various attenuations models. Each of the models require known value of LWC. We can measure weather parameters as density of fog, relative humidity and temperature with using our device [9].

This device has been developed to measure LWC in air, which could indicate fog. Fog has dominant effect on quality of transmission of FSO system and the environment in which FSO system is working. This weather condition composes of water droplets with 100nm in diameter. Examined parameter LWC in this case is measured in unit g/m^3 [10].

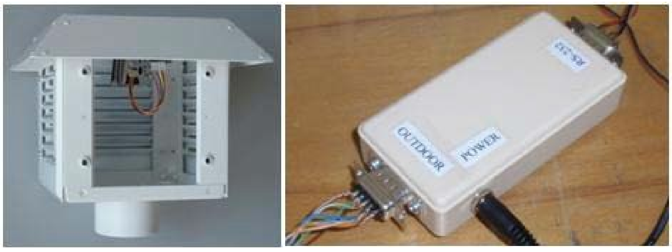


Fig. 5 Hardware configuration of Fog sensor.

In Fig.5 illustrates fog sensor. This sensor consists from two parts. First part is outdoor unit (Fig.5 left) and it is based on sensor which measures water droplets in air. Achieved data are passed to next part of this sensor which converts data gained from sensor to PC. This conversion uses RS 422 and RS 232 lines. The indoor unit is connected to PC by RS 232 line. Measured data are saved for further processing.

V. The Resulting Visibility for the Measured Area

Achieved results from measurement of visibility in our area were processed and evaluated. These results correspond to the area of Košice airport. Results contain data from visibility measurements in periodic intervals. Period of measuring ranges from 1.1.2016 to 31.10.2016. The resulting values for visibility in Year 2016 are displayed Fig.6.

VI. Conclusion

For stable FSO communication system is important secure fast switching to backup line in case when bad weather conditions occur. Higher reliability of FSO communication is secured by good visibility. The resulting values for visibility in Year 2016 show high reliability of FSO system. In our case is distance between communication points in range to 500m. The results show high reliability in this range. Fog in monitored area by results does not have often blackouts caused by fog. Availability for FSO communication system placed in monitored area was calculated as approximately 99,9%. Reliability of FSO system and availability depends on area where FSO system is deployed. When selecting, the place must fulfill condition of good weather conditions.

Acknowledgment

This publication arose thanks to the support of the Operational Programme Research and development for the project "(Centre of Information and Communication Technologies for Knowledge Systems) (ITMS code 26220120020), co-financed by the European Regional Development Fund". This work was supported by research grant KEGA no. 023TUKE-4/2017.

References

[1] M.E. Thomas, D.D. Duncan, “The Infrared and Electro-Optical Systems Handbook” SPIE Press, 1993, vol. 2

[2] P. Liptai, M. Moravec, E. Lumnitzer, K. Lukáčová, “Impact analysis of the electromagnetic fields of transformer stations close to residential buildings” In: SGEM 2014,vol. 1, 2014, pp. 355-360.

[3] Jennifer C. Ricklin, Stephen M. Hammel, Frank D. Eaton, Svetlana L. Lachinova, “Atmospheric channel effects on free-space laser communication”, 2006. doi:10.1007/s10297-005-0056-y.

[4] I. I. Kim, E. Korevaar, “Availability of free space optics (FSO) and hybrid FSO/RF systems”, Proc. of SPIE, vol. 4530, 2001.

[5] P. Ivaniga, T.Ivaniga, "10 Gbps optical line using EDFA for long distance lines", In: Przeglad Elektrotechniczny, ISSN: 0033-2097, doi:10.15199/48.2017.03.45, vol.93, no.3, 2017, pp. 193-196.

[6] P. Liptai, B. Dolník, M. Pavlík, J. Zbojovský, M. Špes “Check measurements of magnetic flux density: Equipment design and the determination of the confidence interval for EFA 300 measuring devices” In: Measurement. Vol. 111 (2017), p. 51-59, ISSN 0263-2241.

[7] M. Pavlík, L. Kruželák, L. Lisoň, M. Mikita, S. Bucko, M. Špes, M. Ivančák, M. Dolník, J. Zbojovský, “The mapping of electromagnetic fields in the environment”, In: *Acta Technica Corviniensis: Bulletin of Engineering.* Vol. 10, no. 2 (2017), pp. 107-110. - ISSN 2067-3809

[8] T. Ivaniga, P. Ivaniga, "Comparison of the optical amplifiers EDFA and SOA based on the BER and Q-factor in C-band", In: Advances in Optical Technologies, ISSN: 1687-6407, doi:10.1155/2017/9053582, 2017, pp.1-9.

[9] M. Tatarko, Ľ. Ovseník, J. Turán, Experimentálne pracovisko pre meranie hustoty hmly, Electrical Engineering and Informatics, proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, 2011 pp. 29-34, ISBN 978-80-553-0611-7

[10] J. Tóth, Ľ. Ovseník, J. Turán, “Free space optics experimental system - long term measurements and analysis”, Acta Electrotechnica et Informatica. 2015. Vol. 15, no 2 (2015), pp.26-30

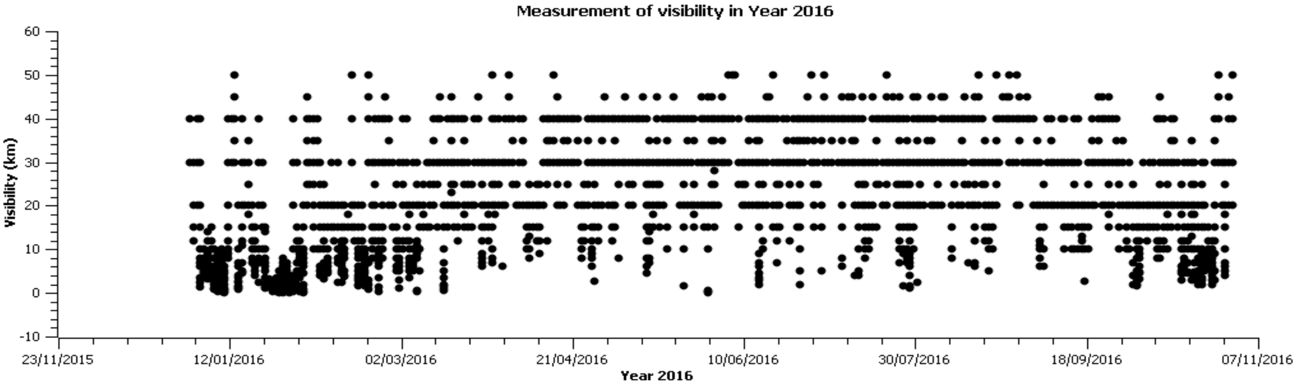


Fig. 6 The resulting values for visibility in Year 2016.

Medical data management

Karol Matiaško, Michal Kvet
Department of Informatics, Faculty of Management Science and Informatics
University of Žilina
Žilina, Slovakia
Karol.Matiasko@fri.uniza.sk, Michal.Kvet@fri.uniza.sk

***Abstract*— Medical data requires high precision and effectivity of the management of measured parameters. Commonly, patients are monitored long time to get the image of the evolution and response to treatment. Sensorial data can be managed and stored using multiple ways, however temporal database allows easy error detection supported by transactions. In this paper, medical temporal management layer is proposed with emphasis on patient monitoring over the time. Effectivity of the solution is provided by multiple layer architecture with data synchronization management and automatic evaluation based on data evolution and significance.**

Keywords—MRI result; medical data; temporality; marker; processing precision bounds

I. Introduction

Techniques for patient examination and treatment options are continually improving. In the recent years, enormous technical development in field of radiology can be perceived. Magnetic resonance imaging (MRI) is one of the most important areas of medicine, in terms of technical capabilities, or in financial terms, however it is still not common part of all hospitals. MRI provides scan method of any part of the human body. It can provide results in form of two-dimensional and three-dimensional images of organs in the human body. Such method has been used in health care since 1980, individual approaches and measurement techniques are improved and offer ability to create high-quality images without the use of radiation [3] [12] [13]. It is not, however, about the production of images, but it is based on measurement, markers and detection techniques of anomalies. Significant aspect is just anomaly location, as well as monitoring aspect. Significant parameter is just the time when the examination is carried out, it supports patient diagnostics, determining and confirming the diagnosis. Imaging methods are also useful during the treatment itself, to monitor particular area of human body, to evaluate the progress of treatment or resistance. Monitoring itself is usually long term process.

In this paper, we will deal with intracranial tumours supported by tumour marker evaluations. In the past, our research was based on anomaly detection and marker value visualization in two and three dimensional environment. An integral part was also module for reducing measurement error. In this paper, we will highlight storage efficiency, methods and techniques for storing MRI results over the time with regards on disc storage capacity as well as particular performance.

Attributes evolve and change their values over the time, therefore each data tuple should be extended by the time definition. But it is not just the values themselves, also structure, precision and characteristics can evolve over the time. The quality of the input data is improved, the equipment is being modernized, so also the system architecture, structure, precision management and characteristics must reflect to it. Therefore, information systems should be aware of them and provide sufficient power for proper, fast and consistent reactions. Data are commonly stored in the database, which forms the core of the information system data management. Conventional approach used today is based on paradigm of storing only current valid data forming actual image [5] [6]. Such approach is, however, not suitable for intelligence aspect of information systems. Therefore, temporal paradigm concept has been proposed. Database management is shifted to the temporal approach, thus, data changes and evolutions are monitored over the time to provide complex information about changes. It contributes to a comprehensive opportunities for decision making either in the context of operational decisions, but also long-term aspect as well.

Temporal database approach is based on temporal paradigm, which extends the definition in time spectrum. Object itself is therefore not defined only by its identifier, but also time validity component is present. Unfortunately, temporal paradigm has not been complexly approved by standardization organizations yet, therefore database systems do not provide sufficient power, individual states cannot be enhanced by the temporal integrity rules [8] (foreign key tuple should cover also time definition of the appropriate primary key tuple), no period data types reflecting time interval positions are available (each state should be defined by only one value for each attribute). As a result, complex solution, individual processes, data modelling and management must be done on your own without complex environment possibilities.

This paper consists of eight sections. Section 2 provides technical background and deals with intracranial tumours and their detection techniques. Section 3 defines marker management provided as result of the magnetic resonance imaging process. Section 4 proposes temporal architecture intended to store result sets. Evaluations and processes are main part of the section 5 supported by physical architecture of the solution defined in section 6. Performance factors, limitations and properties are described in section 7 with emphasis on developed software. Section 8 concludes, summarizes results and proposes new goals.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Intracranial tumours form a complex and diverse group of tumour anomalies, which can differ in location, symptoms, histology and occurrence, which can also depend on age. The most common symptoms are limb movement disorder, numbness, vision, speech or mental changes. Another group of symptoms are resulted from local brain tissue irritation manifested as different types of seizures. Syndrome of increased intracranial pressure is referred to a set of symptoms, which include mainly headache, vomiting and visual disturbances [9] [10] [12].

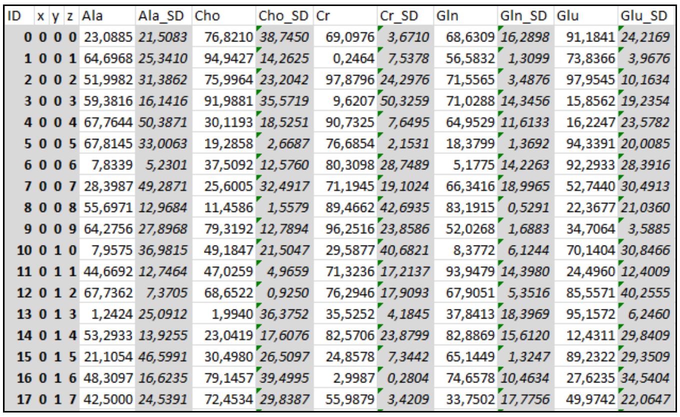
Diagnostics and examination is usually done after detecting symptoms affecting brain functions. There are several imaging methods, like *computer tomography (CT)*, which is based on the principle of X-rays or *positron emission tomography* (PET) based on the principle of monitoring increased sugar metabolism by tumour cells. *Magnetic resonance imaging* (MRI) is specific diagnostic methods used in health care since 1980. It provides two or three dimensional result set of the particular organ based on slices and images. Physical principle is based on measuring changes in the magnetic moments of the nuclei of elements with odd atomic number placed in a strong static magnetic field after application of radiofrequency pulses [3] [4] [11].

II. Markers

Tumour marker is a substance, usually protein, the occurrence of which indicates the presence of cancer. This substance can be almost exclusively produced by the tumour cells, thus the healthy cells do not produce them, but the markers can also be part of the normal cells, the presence of which in abnormal quantity or time also significates the tumour. In general, we can observe the presence of these markers either within the tissue, which can distinguish between normal and damaged tissue or in body fluids (e.g. serum, urine, cerebrospinal fluid), in which these substances are released from the tissues. We use the term marker for a substance that occurs directly in the tumour tissue for this purpose. Markers in this sense may be either protein, or cell surface proteins that are part of the cellular metabolism (such as enzymes, hormones). Most tumours are characterized by the presence of one or several more or less specific markers. An overall picture of the production of these is captured by tracking the presence of multiple markers - although with a lower sensitivity and specificity [10] [12] [14].

III. Position-Temporal architecture

Managing medical data has its specifics, which must be taken into account during the definition. Output of the MRI consists of body slices in two dimensional character, which can be using advanced techniques formed into 3D model. Essential part is just the descriptive data in form of marker values in measured place. Modern devices provide really high quality data, which means, that significant data amount must be evaluated, processed, stored and subsequently retrieved. When we highlight also the necessity for evolution and changes tracking, problem is really severe. Such patients are monitored long time, respective during the whole life. Therefore, it requires sophisticated approach with emphasis on disc storage demands as well as performance. In the first stage of the evolution, result set was stored using standard description files with various formats. Fig. 1 shows the example of the measured markers in the form of data file table output.



ID	x	y	z	Ala	Ala_SD	Cho	Cho_SD	Cr	Cr_SD	Gln	Gln_SD	Glu	Glu_SD
0	0	0	0	23,0885	*21,5083*	76,8210	*38,7450*	69,0976	*3,6710*	68,6309	*16,2898*	91,1841	*24,2169*
1	0	0	1	64,6968	*25,3410*	94,9427	*14,2625*	0,2464	*7,5378*	56,5832	*1,3099*	73,8366	*3,9676*
2	0	0	2	51,9982	*31,3862*	75,9964	*23,2042*	97,8796	*24,2976*	71,5565	*3,4876*	97,9545	*10,1634*
3	0	0	3	59,3816	*16,1416*	91,9881	*35,5719*	9,6207	*50,3259*	71,0288	*14,3456*	15,8562	*19,2354*
4	0	0	4	67,7644	*50,3871*	30,1193	*18,5251*	90,7325	*7,6495*	64,9529	*11,6133*	16,2247	*23,5782*
5	0	0	5	67,8145	*33,0063*	19,2858	*2,6687*	76,6854	*2,1531*	18,3799	*1,3692*	94,3391	*20,0085*
6	0	0	6	7,8339	*5,2301*	37,5092	*12,5760*	80,3098	*28,7489*	5,1775	*14,2263*	92,2933	*28,3916*
7	0	0	7	28,3987	*49,2871*	25,6005	*32,4917*	71,1945	*19,1024*	66,3416	*18,9965*	52,7440	*30,4913*
8	0	0	8	55,6971	*12,9684*	11,4586	*1,5579*	89,4662	*42,6935*	83,1915	*0,5291*	22,3677	*21,0360*
9	0	0	9	64,2756	*27,8968*	79,3192	*12,7894*	96,2516	*23,8586*	52,0268	*1,6883*	34,7064	*3,5885*
10	0	1	0	7,9575	*36,9815*	49,1847	*21,5047*	29,5877	*40,6821*	8,3772	*6,1244*	70,1404	*30,8466*
11	0	1	1	44,6692	*12,7464*	47,0259	*4,9659*	71,3236	*17,2137*	93,9479	*14,3980*	24,4960	*12,4009*
12	0	1	2	67,7362	*7,3705*	68,6522	*0,9250*	76,2946	*17,9093*	67,9051	*5,3516*	85,5571	*40,2555*
13	0	1	3	1,2424	*25,0912*	1,9940	*36,3752*	35,5252	*4,1845*	37,8413	*18,3969*	95,1572	*6,2460*
14	0	1	4	53,2933	*13,9255*	23,0419	*17,6076*	82,5706	*23,8799*	82,8869	*15,6120*	12,4311	*29,8409*
15	0	1	5	21,1054	*46,5991*	30,4980	*26,5097*	24,8578	*7,3442*	65,1449	*1,3247*	89,2322	*29,3509*
16	0	1	6	48,3097	*16,6235*	79,1457	*39,4995*	2,9987	*0,2804*	74,6578	*10,4634*	27,6235	*34,5404*
17	0	1	7	42,5000	*24,5391*	72,4534	*29,8387*	55,9879	*3,4209*	33,7502	*17,7756*	49,9742	*22,0647*

Fig. 1. Input data file – meausrement of the markers

Later, whereas it was really difficult to process several images, database approach has been used. Current solutions for dealing with medical data are mostly based on storing the whole image – result of the examination and processing. In that case, conventional database approach is used - individual examinations can be sequentially numbered and patient information themselves can be one of the standard attribute. Fig. 2 shows the principles of the modelling, one image of the examination is shown. It consists of two parts – personal data information about the patient and the examination results themselves. Structure of any person result set will look like that regardless the number of his examination [7].

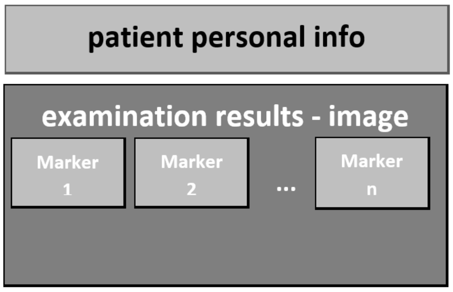


Fig. 2. One patient data result – one image

Main disadvantage of previously described solution is just the comparison and difference evaluation aspect. Individual examinations of the same patient are not grouped together. Moreover, individual anomaly types cannot be grouped, as well. Therefore, in the recent past, particular data were bulked together and loaded to the database. References to the anomaly type or patient data were modelled using pointers. Fig. 3 shows the solution principle. Notice, that each image can be connected to several anomaly types, which can even evolve over the time (in the image 5, new anomaly of the patient 1 was detected). Vice versa, image 3 is not connected to any anomaly, which means, that treatment was successful [1] [7].

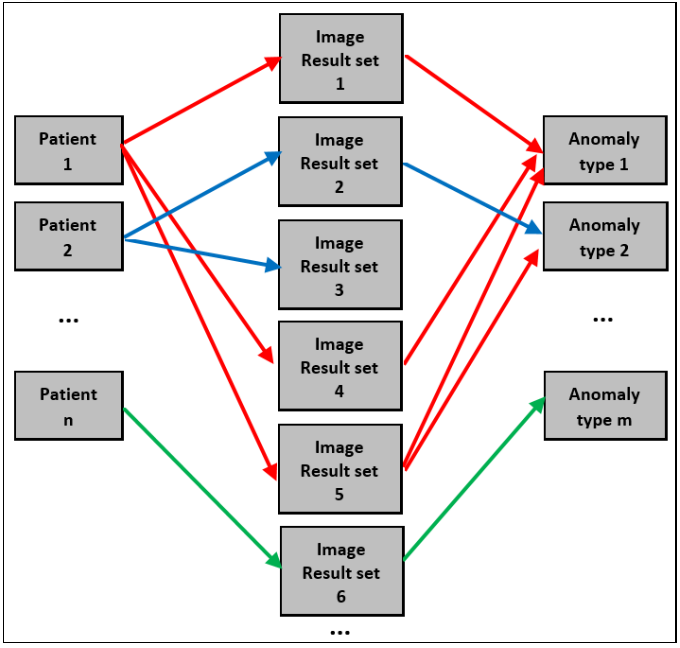


Fig. 3. Image pointers – patient, anomaly type

Evaluation and postprocessing (several images comparison) opportunities are robust. On the other hand, what about the performance and effectivity? Whereas postprocessing is commonly done in the memory, particular examination image must be loaded. Moreover, similar anomalies of the different patients are compared. Naturally, multiple images could not be processed in parallel resulting in increasing resource demands, mostly available memory.

Increasing demands on the resources and capacity have been significant due to improved result quality (modern MRI devices). Therefore, it is necessary to address the way of the data management optimization at different level. As partially mentioned, individual result set consists of several markers, which have been monitored. These set can be adjusted during the treatment. Common part of the processing is just the time and position. Each data tuple has two location attributes – position and time. Rest elements can be changed and can evolve (markers, area of interest, values, resistance, contrast medium, etc.). We have developed two different temporal approaches and we will highlight their benefits and limitations. In the first phase, the whole examination result set (image) was considered as one object. In that case, positional data are changing (individual measured values can be located in the grid), but the time element is the same for the whole image. Therefore the solution leads to the previously defined solutions with no significant performance and effectivity contribution.

IV. Evaluations, Processes

Our second research stream extends previously defined approach and use several techniques and processes to evaluate and compare data over the time. During the first examination, anomalies are detected. Such approach is strict – any suspicious tissue is evaluated as potentially dangerous. Such locations objects can be detected either automatically, but also user (doctor) intervention is available. Afterwards limiting factor values are defined, which border data to be consecutively stored in the database. In principle, any object with abnormal characteristics is monitored automatically. To evaluate impact of treatment, also surroundings is greatly monitored, even with the higher emphasis and precision than the anomaly itself. The reason is just the treatment effect – whether the anomaly is increasing its size or not. The rest part of the organ tissue is also monitored, but with lower emphasis, measurement precision is reduced to such an extent, that it is still possible to recognize any new substandard area. Notice, that these values are evaluated before the storing into the database itself, thus the data amount is significantly reduced. On the other hand, finding any problematic object area automatically invokes increasing accuracy in the area in the consecutive examination. Fig. 4 shows the precision and evaluation level based on defined anomaly. Surrounding width is set based on distinguished anomaly. If it cannot be directly categorized, particular value is undefined and whole spectrum is treated in detail next time. Core of the anomaly is red. In that case, measurement and storing precision is medium. High precision is located in the direct neighborhood of the anomaly (light red color in fig. 4), which sequentially degrades into low precision processed in normal (standard, healthy) tissue. Despite of the fact, that it is delimited by low precision, there is still possibility to locate new nascent anomaly.

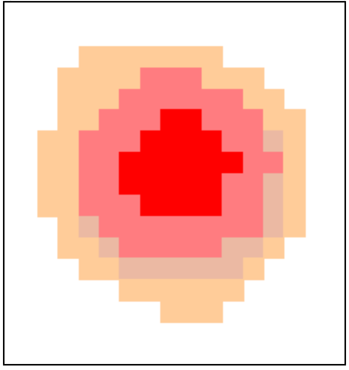


Fig. 4. Detected anomaly

In principle, neighborhood shape can be considered as the circle wrapping the anomaly. It is, however, valid only if the anomaly is strictly bordered. In that case, whole anomaly is considered as one object (cluster) with the same precision and directly callable shape. Another situation occurs, when the object is unstable, broken or creates fractions. In that case neighborhood width is calculated as maximum length. Fig. 5 shows the anomaly tissue, blue color expresses the real neighborhood. It is characterized by irregular shape. In our solution, however, it is always replaced by circle (fig. 6).

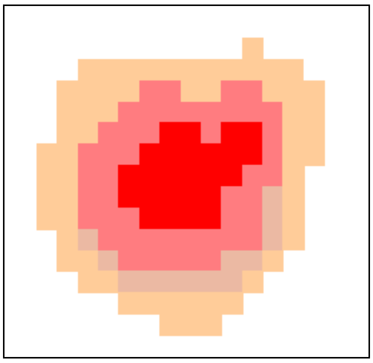


Fig. 5. Detected anomaly – specific shape

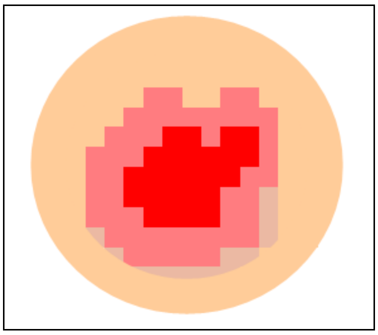


Fig. 6. Surroundings of the detected anomaly – specific shape

V. DATABASE ARCHITECTURE

Examination results have been categorized based on marker and tissue type (anomaly, surroundings, healthy tissue). Each category is defined by separate object – cluster. Surroundings is divided into several categories based on precision, which decreases with the distance from the anomaly itself. Number of surroundings classes is delimited by the *precision_surround* parameter. With an increasing number of classes, also performance increases. Whereas precision differs, each classes and category is processed separately as the attribute (column). Physically, database architecture is covered by the advanced attribute oriented temporal approach. Communication with external sources is done based on *Data Monitor* process, which provides connection between external source (machine, application, etc.) and temporal database, which has three separate parts – temporal management layer, current examination image of particular user and historical images sorted based on the date [2] [8]. Each layer is covered by separate attribute oriented approach – individual attribute is category or its subclass. Fig. 7 shows the architecture.

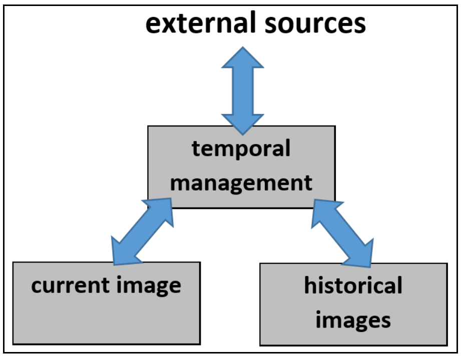
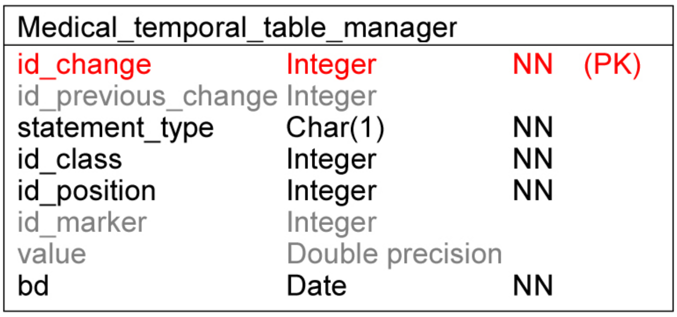


Fig. 7. Database architecture

Core of the temporal model is temporal table with the following attributes (fig. 8):

- *ID_change* – internal number for the examination
- *ID_previous_change* – references the last change of an cluster (anomaly, surroundings class) identified by *ID*. This attribute can also have *NULL* value that means, the data have not been processed yet, so the data were inserted for the first time in past and are still actual.
- *ID_class* – references the cluster class, record of which has been processed.
- *ID_position* - carries the information about the identifier of the row (positional data) that has been changed.
- *ID_marker* – holds the information about the evaluated marker.
- *value* - obtained value referencing particular marker (stored in *ID_marker* attribute).
- *BD* – the begin date of the new state validity of an object – date of the patient examination.



Medical_temporal_table_manager

id_change	Integer	NN	(PK)
id_previous_change	Integer		
statement_type	Char(1)	NN	
id_class	Integer	NN	
id_position	Integer	NN	
id_marker	Integer		
value	Double precision		
bd	Date	NN	

Fig. 8. Temporal table

VI. PERFORMANCE

Experiment results were provided using Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production; PL/SQL Release 11.2.0.1.0 – Production. Parameters of used computer are:

- Processor: Intel Xeon E5620; 2,4GHz (8 cores),
- Operation memory: 16GB,
- HDD: 500GB.

100 patient results have been used, each one was delimited by 10 MRI examinations. Current systems are mostly based on conventional approach. In that case, specific additional

attribute must be used to model primary key (usually defined by the sequence and trigger or by autoincrement column). As the definition is extended by examination time, positional data themselves are not unique. Current approaches are based on packing the whole result image into one block delimited by the time as conventional attribute. The main disadvantage is just the efficiency, when managing duplicate values and comparing positional data result set in the specific time spectrum (timestamp or during defined period). Performance of the temporal system can be characterized in two spectra – size of the whole structure and processing time. In our approach, detection of the positional data duplicate tuples is done automatically, however, improvement is hard to evaluate, whereas it strongly depends on the progress and changes of the anomaly. Significant performance aspect to highlight is just the time of the processing and obtaining results. The aim is to direct the doctor´s attention to the progress and changes of the detected anomalies in time. It is not necessary to monitor complete images of the results, such activity can be shifted to the software detection and management tools. Thanks to that, it reduces time and demand of the processing by the specialists themselves. Another important aspect is the processing time and the process of data retrieval. In this paper, performance definition expresses processing time of the data retrieval process. By using our approach, performance (monitoring evolution over the time) has been improved at rate 35% for only one surroundings class. The worst solution was reached for 100 surroundings classes. However, in comparison with existing approach with constant precision, performance increases at rate 22%. The reason for providing better performance is based on minimizing data to be evaluated as well various precision based on active anomaly. It differs based on the number of classes, because of the recalculation necessity – class can be compared only with those, which have greater precision accuracy. On the other hand, if the number of surrounding classes increases, size requirements are lowered. Fig. 9 shows the dependencies between number classes and size, respectively performance characteristics. By defining 70 groups, these lines influencing performance are crossed.

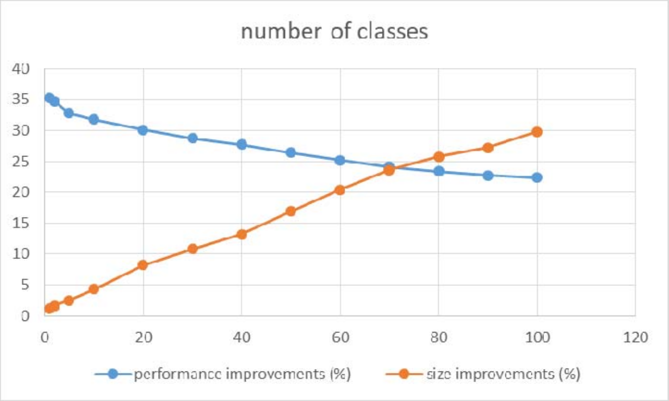


Fig. 9. Performance

VII. CONCLUSIONS

Cancer is one of the most serious medical problems worldwide. Early diagnosis and adequate treatment play a key role in the prognosis of the patient. Nowadays, modern imaging methods are available providing high quality results. In this paper, we deal with magnetic resonance imaging approach, which can provide marker results of the exanimated organ tissue. These marker characteristics are monitored over the time and provide key for consecutive treatment. Important role is therefore effectivity of the marker value monitoring over the time with emphasis on performance and reliability. In our approach, exanimated tissue is divided into several categories, which are characterized by various precision of the stored values. Thanks to that, size can be significantly reduced, but the quality and reliability of the system is maintained. The greatest emphasis of the processing is placed on the anomaly surroundings, which is the most important for treatment effect evaluation with clear monitoring methods. Data are physically stored in extended temporal database architecture, based on attribute oriented granularity, which is delimited by the processed precision area.

In the future, we will deal with extension of the anomaly detection and complex management of the class number with emphasis on the global performance.

ACKNOWLEDGMENT

This publication is the result of the project implementation: *Centre of excellence for systems and services of intelligent transport*, ITMS 26220120028 supported by the Research & Development Operational Programme funded by the ERDF and *Centre of excellence for systems and services of intelligent transport II.,* ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

This paper is also supported by the following project: "*Creating a new diagnostic algorithm for selected cancers*," ITMS project code: 26220220022 co-financed by the EU and the European Regional Development Fund.

"PODPORUJEME VÝSKUMNÉ AKTIVITY NA SLOVENSKU

PROJEKT JE SPOLUFINANCOVANÝ ZO ZDROJOV EÚ

REFERENCES

[1] K. Ahsan, P. Vijay. “Temporal Databases: Information Systems”, Booktango, 2014.

[2] C. J. Date, N. Lorentzos, H. Darwen. “Time and Relational Theory : Temporal Databases in the Relational Model and SQL”, Morgan Kaufmann, 2015.

[3] Hornak, J.: The Basics of MRI, Interactive Learning Software, 2008.

[4] James, S.: 3D Graphics with XNA Game Studio 4.0. UK: Packt Publishing Ltd., 2010. ISBN-10: 1-849-69004-9,

[5] T. Johnston and R. Weis, “Managing Time in Relational Databases”, Morgan Kaufmann, 2010.

[6] M. Kvet, K. Matiaško, “Transaction Management in Temporal System”, 2014. IEEE conference CISTI 2014, 18.6. – 21.6.2014, pp. 868-873

[7] M. Kvet and K. Matiaško, “Uni-temporal modelling extension at the object vs. attribute level”, IEEE conference UKSim, 20.11 – 22. 11.2014, , pp. 6-11, 2013.

[8] M. Kvet and K. Matiaško, „Temporal transaction integrity constraints Management“, Cluster Computig, Volume 20, Issue 1, pp. 673–688

[9] Mešina, J.: 3D vizualizácia mozgu človeka (diploma thesis), 2012, Žilina

[10] Nekula, J.: Radiologie, Univerzita Palackého, 2005. ISBN: 8024410117

[11] Pianykh, O. : Digital Imaging and Communications in Medicine, Springer, 2008. ISBN: 978-3-540-74570-9

[12] Uecker M.: Autocalibrating and calibrationless parallel magnetic resonance imaging as a bilinear inverse problem, IEEE Conference Publications, 2017, pp. 111-113.

[13] Xing F.: Phase Vector Incompressible Registration Algorithm (PVIRA) for Motion Estimation from Tagged Magnetic Resonance Images, IEEE Transactions on Medical Imaging, Volume PP, Issue 99, 2017.

[14] Zeman, M.: Speciální chirurgie, Galén, 2006. ISBN: 8072622609

NFC/RFID technology using Raspberry Pi as platform used in Smart Home project

Juraj Mihaľov
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Slovak Republic,
juraj.mihalov@tuke.sk

Michal Hulič
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Slovak Republic,
michal.hulic@tuke.sk

Abstract— **This article deals with possibilities of using the Raspberry Pi microcomputer device as a learning platform for students, with history of development of mentioned device and comparison with available competition, characteristics of previous versions of the microcomputer for learning purposes ideal for students to gain an experience with real life solution described in this paper. Our Smart Home project contain different motion sensors, an ordinary sensors and we want to integrate NFC/RFID technology too. NFC/RFID technology can be used as registration platform for members of Smart Home, also this technology can be used with security devices such as cameras, sensors and other devices which they can measure basic parameters of the building or energy devices and using PubNub, we can power bidirectional communication between the devices at home and our browser or mobile device.**

Keywords— NFC, Raspberry Pi 3, RFID, Smart Home**.**

I. Introduction

This paper shows part of the process of building Smart home system with used hardware parts to make registration for members of Smart Home. In many services which provides Smart Home are still an obsolete system where participants have to physically notify their attendance long time ago before their presence can be allowed [1]. Such system is unbearable so that is the base of idea for this project. There is also another fact. Smart Home we see as project with dynamic evolution of members. All the people who want to be a member of Smart Home, have to be registered. Registration is placed at first visit of the Smart Home [2]. Members have different privileges. Members have two options. If they have an NFC (Near Field Communication) device as smartphone, wearable smart watch or other device that can communicate by NFC or they choose the other option where they receive RFID chip or key ring with RFID chip [3]. NFC is a way for two devices to communicate very close to each other. Sort of like a very short-range Bluetooth that doesn't require authentication [4]. It is an extension of RFID, so anything you can do with RFID you can do with NFC [5]. You can do more stuff with NFC as well, such as communicate with cell phones bi-directionally [6]. Also, the Arduino is compatible to Raspberry Pi for extension its features. That features are really suitable for smart home applications, for cooperation with Kinect for 3D sampling of the space of the room or the exterior [7]. Also the other supplements are able to connect to Arduino with cooperation with Raspberry Pi that opens endless possibilities for applications which includes hardware [8]. Raspberry Pi has one big advantage that has GUI (graphical user interface) which is much more usable for creating applications for control the hardware such as sensors Leap motion and Myo Armband [9]. This information is sent from one device to the other, without having to open any ports on the devices, through the network they are on [10]. PubNub is the key communication component between all the things in the Internet of Things, and in this case, our Pi house [11]. The IoT things are the most trending technology today, in our case we used Raspberry Pi 3. It is a very simple concept where devices in our home or wherever they are, have the capability to communicate with each other via the internet [12]. Usually sensors are used with this technology to pass data to the internet. We can imagine a sensor installed in our house which uploads data like temperature, humidity, etc. to the internet, and this data are visible at simple web UI [13].

II. Main Components

A. Raspberry Pi 3 and relevant parts
B. PN532 NFC/RFID controller
C. Software part – Setting up sensors and UI
D. Server side
E. UI for data visualization

A. Raspberry Pi 3 and relevant parts

Due to microcomputer Raspberry Pi with its small dimensions does not look like a full-valued computer, with its functionality has no difference with personal computer. The device was developed to promote the teaching of basic computer science in schools and developing countries. The raspberry Pi 3 model B contains quad-core Cortex-A53 processor with architecture ARMv8-A (64/32 bit version), which is described as 10 times the performance of the first model Raspberry Pi 1. That was suggested to be highly

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

dependent upon task threading and instruction set use. Raspberry Pi 3 is considered as 80 percent faster then Raspberry Pi 2 in parallelized tasks. The model 3 used in this paper also includes 1 GB SDRAM memory that is shared with GPU. The GPU Broadcom VideoCore supports OpenGL library and has many resolution options on different frame rate from VGA resolution 640x480 to WUXGA resolution to 1920x1200. The device even supports 4K resolution 3840x2160 but only at 15 Hz frame rate that is too low for convincing video. But mentioned specification is highly sufficient for this application. The model 3 is able connect to network using 100 Mbit Ethernet port and also 802.11n wireless and Bluetooth standard 4.1. One of the main part is low level peripheral 40 pins 28x GPIO, I2C, SPI, UART that provides ability to connect expansion card to increase the ability of connecting peripheral electronics such as in our case RFID/NFC chips to connect with other devices. The device supports much operating systems such as CentOS, Raspbin, Windows 10 IoT, Debian etc. With its dimensions its really suitable for smart home application, because the parameters it supports its totally sufficient to support all the functionalities we need to build smart home application with no need to use the personal computer.

Due to presence of interfaces such as USB, HDMI etc. can connect peripheral devices similar as personal computer and has similar functionalities. The main part in our project is that device can be programmable that has an excellent value for master of the Smart Home.

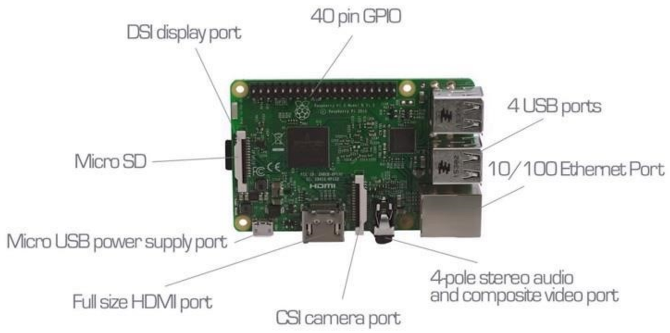


Figure 1 Microcomputer Raspberry Pi 3, Model B [14]

From the wide options of prototyping platforms is Raspberry Pi 3 the most suitable for our project. In our solution we were thinking about platforms as Asus Thinker Board, Pine A64+ and Banana Pi 3 [15]. The basic parameters of microcomputer Raspberry Pi 3 are shown in the table 1 below. Project has mandatory parts without the presence has no sense to build this project. Ability to connect display has great option for all member of Smart Home. That means that device can be controlled by this peripheral and it is suitable for many members. With the display is also in pack adapter board, DSI cable and other hardware components to mount to display to prepared container [16].

Table 1 Basic parameters of Raspberry Pi 3 model B [17]

4-core, 64 bit processor, 1,2Ghz, ARM Cortex A53	
1GB RAM	400Mhz VideoCore IV GPU
Bluetooth 4.1 +BLE	WiFi 802.11n
Ethernet port	Full HDMI port
3.5mm audio jack	CSI camera port
DSI display port	MicroSD card port
VideoCore IV 3D	2.5A input current
4x USB 2.0 port	40 GPIO pins

B. PN532 NFC/RFID controller

RFID is the method of uniquely identifying items using electromagnetic fields for identifying and tracking tags. RFID system comprises reader, tag and antenna. Reader sends an interrogating signal to the tag via antenna and tag responds with its unique information. The controllers are divided in two main group. Passive and Active. Passive are powered by electromagnetic energy transmitted from the RFID reader. Active RFID tags contain their own power source giving them ability to broadcast with a long range [18].

Passive RFID tags operate at:

- Low Frequency 125 – 134 kHz,
- High Frequency – 13.56 MHz,
- Ultra-High Frequency 856 – 960 MHz.

Differences between active and passive RFID are shown in table 2.

Table 2 Difference between active and passive RFID tags [15]

	Active RFID	Passive RFID
Distance	30 meters/100feet	6 meters/20 feet
Power Source	Internal Battery powered	External- Relies a reader
Data Storage	128kb read/write	128b read/write
Tag Expiration	5-10 years, dependent on the battery's life.	Often longer than a lifetime depending on the environment.
Size	Large enough to accommodate the battery.	As small as a microchip and as large as a paperback book.
Advantages	Reads long distances. Highest data bandwidth. Able to initiate communication. Tag must be replaced when battery dies.	Longer lasting, tag life doesn't depend on battery. Tags are inexpensive. Small tag size accommodates range of assets and is easy to conceal. More resistant to physical damage or harsh environment.
Disadvantages	Tags are costly. Cannot function without battery. The tags are large in size, not suitable for smaller assets.	Communication depends on the antenna size and shape. Read range is limited. Difficulty reading through metal or liquid.
Cost	Around 15-20 per tag	Around 0.1-0.2 per tag

NFC devices operate at the same frequency as high frequency RFID, 13.56 MHz. PN532 is NFC chip is embedded in almost every smartphone or device that does NFC. Identification is based on contactless communication between electromagnetic wires reader and chip which is bidirectional. RFID and NFC has one big advantage over bar codes that communication may take place as wireless and can be scanned from short distance. NFC technology is much safer than RFID because the range of NFC is only few centimeters. NFC is may act as reader or tag. The next advantage is that communication is based on peer-to-peer protocols. For our solution is necessary to obtain parts designated for Raspberry Pi:

- HDMI cable,
- Power supply,
- Necessary peripherals (mouse, keyboard),
- Monitor,
- MicroSDHC card with pre-installed operating system.

RFID and NFC logos are similar and both of them are shown at figure 2. In practice are used as sticks to the tag.

Figure 2 RFID and NFC logos [20]

In comparison technologies RFID and NFC with technology of Bluetooth is the main advantage that device has no need to pair. With usage with platform Raspberry is the most suitable option extendible module with chipset PN532 compatible with microcontroller Raspberry Pi 3 produced for example by companies Adafruit or Itead [21] [22]. The PN532 is the most popular NFC chip, and is embedded in pretty much every phone or device that does NFC. It can pretty much do it all, such as read and write to tags and cards, communicate with phones (say for payment processing), and 'act' like a NFC tag. It is compatible with all RFID tags and readers. PN532 NFC module is equipped with double-row pins, which can be connected directly to the Raspberry Pi via connection cable and then drive the module for non-contact near field communication operations. Module PN532 used in our solution is shown at figure 3.

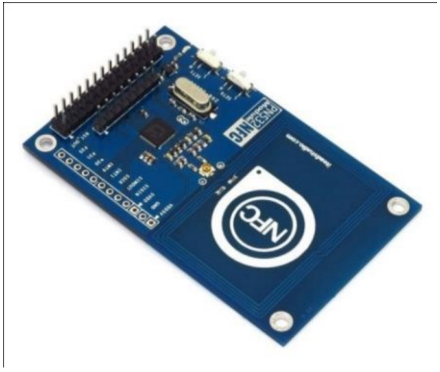


Figure 3 Module PN532 NFC/RFID [23]

C. Setting up

Module PN532 is able to work with Raspberry Pi platform SPI and I2C bus. For correct connection it is necessary to connect pins according the table 3. Registration of the members was done by the database stored in Raspberry Pi using python code for reading from RFID reader. To store the user was use 6 hex numbers with 6 digits, thanks to we can store up to 16 million users. The value which is bigger that 6 digits is error and cannot be stored in database. The first step is to read the RFID chip that connected to PN532. The second step is to pick a block type. The third step is to confirmation writing the block type. And confirmation is the last step. The user ID is assigned to the relevant card where the user can be identified.

Table 3 Raspberry Pi 3 and PN532 NFC/RFID connection

Raspberry Pi 3 Model B	PN532 NFC/RFID module
5V	5V
Ground	Gnd
GPIO10 (SPI_MOSI)	MO/SDA/TX
GPIO09 (SPI_MISO)	MI
GPIO11 (SPI_CLK)	SCK
GPIO08 (SPI_CE0_N)	NSS/SCL/RX

The reading from the card is done by another python file to scan the chip from where the data are send to database using PN532 NFC RFID card reader at specific frequency 13.56 MHz. By this project was also used a Libnfc library that is a public platform independent Near Field Communication library. Due to fact that library is open source it provides complete transparency and royalty free use for everyone. For usage it has to be downloaded and configured for proper functioning. Due to various operating system has to be downloaded by supported commands and configured. This library is first library at low level for NFC SDK and programmer's API. Library also supports various modulations that can work in mode as target and as initiator. This technology is great option not only for Smart Home, also for smart cities thank for possibility of the payment or registration of the cars or people moving around the city. Smart Home option is available to control the devices and environment around the house. The example of the possibility for Smart Home using is smart card door locker. With technology NFC we are able to make a database of persons who are permitted to enter the room using technology as Raspberry Pi cooperating with Arduino and NFC/RFID PN532 card reader. The mechanical parts are necessary to involve due to need opening or closing the door. For the work with records is uses application the spring framework software. It can locally connect to MySQL database server that is installed on lipsberry pi. A database for storing records is created on the server.

The basic principle of gathering and sending information is in following two steps:

- Read the sensor values such as temperature and humidity on a real time web UI,
- Monitor the same devices, and send control messages to them, to turn them on/off, to check their state and to take any action on them [22].

Step 1: Reading the values from sensor

The code makes the Pi read the humidity and temperature values from the sensor and print it out. At this point, our sensor is collecting temperature and humidity data, but not doing anything with it. We want to stream this data to a web browser and display it in real time, reflecting changes as they happen.

Step 2: Install PubNub service
Install PubNub:

```
pi@raspberrypi ~$ sudo pip install pubnub
```

After make sure we import the right libraries needed for this program. We initialized an instance of PubNub by inputting us publish/subscribe keys. Our channel is where we publish the message to, so choose a unique name for that. In order to view these readings on our web UI, we need to publish them on a specific channel using PubNub. The browser subscribes to the same channel, and hence receive the message. Here we are formatting the data to two decimal places, forming a message to be sent over PubNub, and finally publish the message. Before we can start coding with PubNub, we need to make sure all the libraries we need are installed and up to date. We opened up terminal, then we typed the following commands to update and install the necessary libraries.

```
pi@raspberrypi ~$ Sudo apt-get update
pi@raspberrypi ~$ sudo apt-get install
python-dev python-pip pi@raspberrypi ~$
sudo pip install pubnub
```

To start using PubNub, we initialized it using publish and subscribe keys. The Raspberry Pi smart home is all about IoT, and IoT is all about the devices communicating with each other in real time. PubNub powers that communication between devices. Whether it's a mobile device or a web browser talking to embedded devices, sensors or any other device, PubNub glues them together. In this specific example, we use a web browser to communicate with the sensors and the Raspberry Pi, to collect and receive temperature and humidity values and NFC/RFID values to identify the user of the home. The sensor measures them, and sends it back over PubNub, allowing you to visualize it on your browser in real time.

D. Server side

Also the server side of the system has to be set up. If we would use our own domain, we can use the file manager service to drop these files into server and can be used to control IoT systems from anywhere around the world. The files to be put on our server are main.html. The program raspbi.py is the python program which is to be copied to raspberry pi. The HTML file consists of a basic UI with two buttons. On button press, it triggers a PHP program which writes a string to a .txt file depending on the button pressed. The .txt file stores information of the last button state.

Due to we don't have a domain, we can simulate a domain in local network or Wifi using a service called xampp. It is very simple solution. Using xampp we can use our local PC as a local web server. However, the IoT systems get confined to a control range limited to wifi network. We dropped the HTML, PHP and .txt files into the /htdocs/xampp directory [21].

E. UI for data visualisation

Our web UI is the control station for the Raspberry Pi smart home. This is where we see our temperature and humidity readings, can check the status of connected devices, and can control and trigger action, all in real time also we can see the presence of the connected persons. In future we can make the profile for the users based on NFC/RFID technology and the system preferences for specified user. The UI updates as the data changes [20].

III. Conclusion

The presented project shows that microcomputers such as Raspberry Pi are suitable option as part of Smart Home project. It is cheap and functional way how to control different peripherals. Function of Raspberry in our project is NFC and RFID authentication using mentioned technologies for Smart Home. Members can be easy added to database and administrator can manage their privileges. Our first application to Smart Home is NFC/RFID controlled door lock which is main part of Smart Home in our view. The next step was implementation of web User Interface, thanks to we can control the Raspberry Pi and gain information from sensors in real time using network connection. In future we will make Raspberry, NFC and motion sensors work together in one secure part of Smart Home project. This will include the cameras such as Kinect, NFC/RFID readers to identify the user who will join the system, in our conditions it means to enter the house or even specific room, the database for identification and authentication occupants, their permission to access the system and execute the specific acts. It may also include two factor authentication with NFC reader and the camera identification such as Kinect for identification person with its permissions. For this requirements are also suitable Raspberry Pi 3 with its parameters to satisfy the difficulty of the extension for smart home project.

Acknowledgment

This work was supported by KEGA Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under Grant No. 077TUKE-4/2015 „Promoting the interconnection of Computer and Software Engineering using the KPIkit“.

This work was supported by KEGA Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under Grant No. 003TUKE-4/2017 Implementation of Modern Methods and Education Forms in the Area of Security of Information and Communication Technologies towards Requirements of Labour Market. This support is very gratefully acknowledged.

References

[1] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, E. Jansen, The Gator Tech smart house: a programmable pervasive space, Computer 38 (2005) 50–60.

[2] O. Brdiczka, M. Langet, J. Maisonnasse, J. Crowley, Detecting human behavior models from multimodal observation in a smart home, IEEE Trans. Autom. Sci. Eng. 6 (4) (2009) 588–597.

[3] A. Rahul, G. Krishnan, U. Krishnan H, and I. Sethuraman Rao, "NEAR FIELD COMMUNICATION (NFC) TECHNOLOGY: A SURVEY" International Journal on Cybernetics & Informatics (IJCI) Vol. 4, No. 2, April 2015.

[4] V. Coskun, B. Ozdenizci, K. Ok., "A Survey on Near Field Communication (NFC) Technology", Wireless Personal Communications: An International Journal, Volume 71, Issue 3, August 2013, pp 2259-2294, doi>10.1007/s11277-012-0935-5.

[5] F. Yanhui, "Internet of Things Application of RFID Technology in Warehouse Management", Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 23 June 2014, ISBN: 978-07695-5122-7, DOI: 10.1109/IMCCC.2013.375.

[6] A. Doaa, A. Gaber, A. Aleem, "Near-Field Communication Technology and Its Impact in Smart University and Digital Library: Comprehensive Study", Journal of Library and Information Sciences December 2015, Vol. 3, No. 2, pp. 43-77, ISSN 2374-2372 (Print) 2374-2364 (Online), DOI: 10.15640/jlis.v3n2a4.

[7] SUEHLE, R. - CALLAWAY, T., 2014. Raspberry Pi Hacks. United States of America, December 2014, 2.vydanie, 247 s. ISBN 978-1-44936234-8.

[8] Ieleni Yagna, „Bomb Defusing Robot Controlled by Gestures with Arduino and Leap Motion“, International Journal od Scientific Engeneering and Technology Research, vol. 04, issue 24, ISSN 23198885.

[9] Augsten, T., Kaefer, K., Meusel, R., Fetzer, C., Kanitz, D., Stoff, T., Becker, T., Holz, C., Baudisch, P.: Multitoe: high-precision interaction with back-projected floors based on high-resolution multi-touch input. In: Proc. UIST 2010, pp. 209–218. ACM (2010).

[10] P. Bhaskar Rao, S.K. Uma, "Raspberry Pi Home automation with wireless sensors using smart phone", International Journal of Computer Science and Mobile Computing. 2015.

[11] S. Goodwin., "Smart Home Automation with Linux and Raspberry Pi", Springer online, 2013., ISBN-13 (electronic): 978-1-4302-5888-9.

[12] V. Vujović, M. Maksimović., "Raspberry Pi as a Sensor Web node for home automation"., Computers & Electrical Engineering, Volume 44, May 2015, Pages 153-171.

[13] E. Upton and G. Halfacree, "Raspberry Pi User Guide", John Wiley & Sons, 2014., ISBN: 978-1-118-92166-1.

[14] Raspberry Pi 3 Model B, https://2.bp.blogspot.com/-GOPkwrpFMBk/VtRkngTJWEI/AAAAAAAAm_U/AFN47w1rb2s/s1600/raspberry-pi-3-microcomputer.png.

[15] ZDNET, "10 Alternatives to the raspberry pi[online 15.9.2017], www.zdnet.com/pictures/10-alternatives-to-the-raspberry-pi.

[16] RASPBERRY PI FOUNDATION, 2006. Raspberry Pi Products [online 14.9.2017]. https://www.raspberrypi.org/products.

[17] RASPBERRY PI 3 IS OUT NOW! SPECS, BENCHMARKS & MORE, [online 15.9.2017] https://www.raspberrypi.org/magpi/raspberry-pi-3specs-benchmarks.

[18] Rochman Saefulloh Basyari, Surya Michrandi Nasution, Burhanuddin Dirgantara, “Implementation of host card emulation mode over Android smartphone as alternative ISO 14443A for Arduino NFC shield”, Control, Electronics, Renewable Energy and Communications (ICCEREC), 2015 International Conference, 30 November 2015, DOI: 10.1109/ICCEREC.2015.7337036.

[19] R. Tesoriero, J. Gallud, M. Lozano, “Using active and passive RFID technology to support indoor location-aware systems”, Published in: IEEE Transactions on Consumer Electronics (Volume: 54, Issue: 2, May 2008), DOI: 10.1109/TCE.2008.4560133.

[20] Corerfid.com, “NFC and RFID logo”, Available at: http://www.corerfid.com/wp-content/uploads/2015/08/nfc-rfid-logo.jpg.

[21] Adafruit company, online 16.9.2017, “https://www.adafruit.com/”.

[22] Itead company, online 16.9.2017, “https://www.itead.cc/”.

[23] NFC/RFID PN 532 module, online 16.9.2017, “http://tinkersphere.com/2207-thickbox_default/pn532-nfc-rfidmodule.jpg”.

Dataflow Processing of Matrices and Vectors: Experimental Analysis

Jurij Mihelič
Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
jurij.mihelic@fri.uni-lj.si

Uroš Čibej
Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
uros.cibej@fri.uni-lj.si

***Abstract*—In this paper we focus on algorithms that extend classical control-flow computation with a dataflow computing paradigm. In particular, our goal is to experimentally explore various dataflow techniques and features, which enable the acceleration of algorithms. One of the most important challenges in designing a dataflow algorithm is to determine the data choreography resulting the running-time performance as good as possible. In the paper, our subject of interest are the algorithms that use matrices and vectors as an underlaying data structure. We discuss several variants of data choreographies for streaming of matrices. We also present the results of experimental evaluation of using these choreographies on several matrix-related problems.**

***Index Terms*—dataflow, choreography, matrix, algorithm, experiment, evaluation**

I. Introduction

Modern day computers are dominated by the von Neumann architecture, even though several alternatives exists [1]. Dataflow architecture is one such alternative [2], once viewed as a competitor to the von Neumann architecture, but now considered more as a complementary one.

Recent technological advances, driven mainly by Maxeler [3], brought the dataflow paradigm back to life by making it not only competitive with the control-flow processors, but overtaking them in many aspects [4]. For example, the possibility to manipulate huge amounts of data while at the same time consuming significantly less energy than comparable solutions based on control-flow processors [5], [6].

Unfortunately, the speedups offered by the dataflow approach are not straightforward: the algorithms have to be carefully re-engineered for most of the current algorithms are tailored specifically for the control-flow architecture. Several successful examples, such as [7]–[9], of such engineering already exists in various application domains (mostly numerical computation).

With this paper we try to shift focus to domains where dataflow computation may not be deemed so successful due to several reasons such as low data reuse and small loop bodies, which basically results in a less computation per an input element. We strive to explore various dataflow techniques using algorithm engineering [10], experimental evaluation [11], and good practices [12] in order to show their practical applicability. In particular, our subject of interest are the algorithms that use matrices and vectors as an underlaying data structure, for which we examine several data choreographies for data streaming, and, finally, we experimentally evaluate them and present the results.

In the next section we present stream processing of matrix-based data, where we focus on several possible data choreographies. Afterwards we present the results of the experimental comparison of selected choreographies, when used in algorithms for solving various matrix-based problems.

II. Stream Processing

The dataflow architecture is based on the stream processing paradigm [2], [13]. In particular, the dataflow process has one or more inputs as well as outputs, where each is a stream of data (i.e., characters, fixed- or floating-point numbers, etc.). In this section, we briefly present how vectors and matrices are stored in the main memory of the computer followed by a discussion on various techniques how they can be efficiently streamed to the dataflow engine. While presenting we base our discussion mainly on the matrix-vector and matrix-matrix multiplication problems.

A. Computer Representation

In mathematics the representation of vectors and matrices is not important (i.e., they are conceptual objects): one just refers to its elements using the subscript indices, i.e., v_i refers to the i-the element of the vector v while $m_{i,j}$ refers to the element in the i-th row and j-th column of the matrix m. On the contrary, in the computer science, the representation is of uttermost importance for it may have a profound effect on the algorithm performance due to the specifics of a particular computer architecture [10], [11].

Vectors are most often represented as a sequence (i.e., an array) of elements. In particular, if A denotes a array, then simply $a_i = A[i]$. (In all the examples we consider that indices start from 0.)

Polynomials are another mathematical object which can be efficiently represented with a vector: the coefficients (including zeros) are listed in a vector in the increasing order of term

This work was partially supported by the Slovenian Research Agency and the projects "P2-0095 Parallel and distributed systems" and "N2-0053 Graph Optimisation and Big Data".

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

degrees. When there are many zero coefficients, i.e. also called *sparse* polynomials, instead both degrees and coefficients are listed either using two vectors of a vector of pairs.

There exists two main representations for the matrices: the *row-major* and *column-major* order [14]. In both, the elements are listed consecutively as they appear in the matrix, but in the former they follow the rows from the top to the bottom as well as from the left to the right inside each row, while in the latter they follow the columns from the left to the right as well as from the top to the bottom inside each row.

Let A denote the (zero-indexed) array to store the matrix; thus, we have $a_{i,j} = A[i \cdot n + j]$ in the row-major representation, and $a_{i,j} = A[i + j \cdot n]$ in the column-major representation.

It is straightforward to convert a matrix between the two representations using the matrix transposition operation. However, the in-place transposition for non-square matrices may be much more elaborate. If A is a matrix then A^T denotes a transposition of A. If A is represented in row-major order then A^T is its column-major representation and vice versa.

B. Rowwise Processing

Assuming row-major representation in the main computer memory, the most straightforward from a user's point of view is to stream the matrix elements as stored in the memory. This results in the data choreography as depicted in Fig. 2 a).

A disadvantage of such choreography is that it often causes a *static* loop (depending only on the computation) in the resulting dataflow graph, and, hence, causing an immediate slowdown (proportional to the size of the loop) of the whole kernel.

For example, when multiplying a matrix with a vector, a dot products of row-vectors with a given vector are computed. Here, each addition operation in the dot product has to wait for its result to become available to be reused as an input. See also Fig. 1 for a simplified representation of the dataflow graph corresponding to a calculation of a dot product.

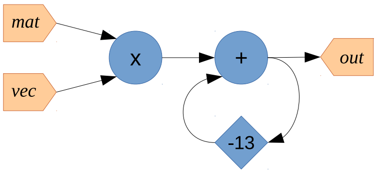


Fig. 1. A loop in the dataflow graph

A common technique for creating dataflow implementations is, similarly to the thread-based control flow parallelization, to replicate a single stream computation (also called a *pipe*) within a kernel in order to process several elements at a time: here, p pipes have a p-fold potential increase in the performance. The replication may mitigate a slowdown caused by a loop in the dataflow graph. However, other factors, such as maximum bandwidth of PCIe (Peripheral Component Interconnect Express, [16]) bus may also effect the performance.

Let us now briefly discuss two options for parallelization of rowwise processing, namely we discuss piped based replication in the *rowwise* and *columnwise* direction. Both options are graphically presented in Fig. 2 b) and c), respectively. Notice that, using p pipes, the general processing order is still rowwise, i.e., the next p elements are taken in the rowwise direction; however, either p elements in the corresponding row (in the former) or column (in the latter) of the matrix are processed at a time. The columnwise direction requires a rearrangement of the matrix elements as they are stored in row-major order in the main memory.

C. Columnwise Processing

Often a loop in the dataflow graph, caused by a dependence of consecutive operations, can be completely mitigated, e.g. in matrix-vector multiplication, using a columnwise processing, where the elements are accessed sequentially from the top to the bottom by columns, starting in the top-left corner and ending in the bottom-right corner. Assuming row-major representation in the main computer memory, the input matrix must be transposed before it is fed to the kernel. The data choreography is depicted in Fig. 2 d).

Obviously, the change in the choreography requires the change in the dataflow kernel. In particular, algorithms such as matrix-vector multiplication need to store the partial results of processing the previous column (i.e., accumulated), when processing the current one. Fortunately, in columnwise processing, these accumulated values are already available when needed since the latency of the required operations is (usually) lower then the column size. Such dependence of data is called a *dynamic* loop in the dataflow terminology for it depends on the input data, e.g., the matrix size.

Again we explore the rowwise and columnwise possibility of computation replication with pipes; see also Fig. 2 e) and f) for the respective data choreographies. Here, the first processes the columns of width p (the matrix has to be preprocessed to be streamed in the corresponding order) while the second takes p elements at a time from the current column.

D. Stripped Processing

Both techniques presented in the previous two sections have some drawbacks. The rowwise choreography contains a static loop to overcome for the latency of the addition operation, while the columnwise choreography produces a long dynamic loop to store intermediate results. Hence, the former introduces a multi-fold kernel slowdown, and the latter consumes significant amount of FPGA resources. To mitigate these two issues, we present another technique which is a combination of both techniques.

The main idea is to divide a matrix into horizontal *stripes*, which are furthermore processed in a columnwise fashion. See Fig. 2 g) for a graphical representation of the choreography. Thus, the speed of columnwise technique is retained, while the resource consumption is significantly reduced. Observe also, that analogous choreography with vertical stripes is also an option; see also Fig. 2 d) for an example.

Denote with s the stripe width, i.e., the number of elements from a particular column. Since the stripe length is n, we have sn elements in total for each stripe. There are n/s stripes and

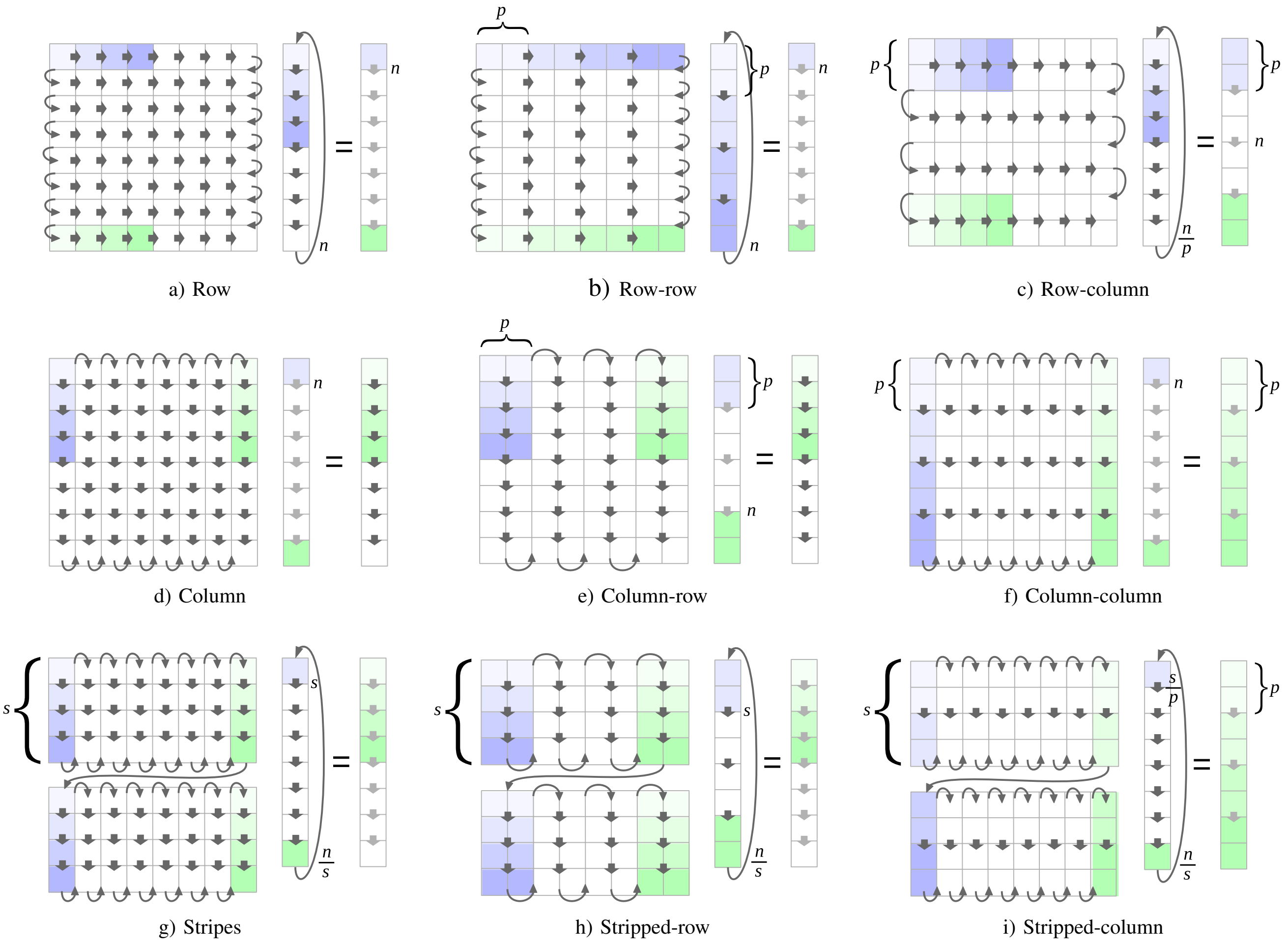


Fig. 2. Various data choreographies: rowwise (top row), columnwise (middle row), and stripes (bottom row), and replication of stream computation based od pipe parallelism: no replication (left column), rowwise replication (middle column), columnwise replication (right column).

for every stripe the whole vector must be streamed. For the fast re-streaming of the vector the fast memory of DFE unit may be used.

Processing of each stripe may be additionally parallelized using pipes. See Fig. 2 h) and i) for the two common techniques, both of which are based on the same ideas as already discussed in the previous two sections. Obviously, the parallelization technique is applied to each stripe separately. Finally, notice that, in rowwise parallelization $s \leq n$ and $p \leq n$ while in columnwise parallelization $p \leq s \leq n$ must hold.

E. Blocks

Now we present a choreography which is suitable for problems such as matrix multiplication. First let us explain what a block matrix is. It is a matrix that is interpreted as having been composed of (non-overlapping) submatrices, called *blocks*. One can also visualize such matrix with a collection of vertical and horizontal lines that partition it into a collection of smaller submatrices. In general, blocks can be of different sizes, and, thus, there are several possible ways to interpret a particular matrix as a block matrix.

Consider matrices $A = [a_{i,j}]$ of dimension $m \times l$ and $B = [b_{i,j}]$ of dimension $l \times n$. Now, a block matrix A with q row partitions and s column partitions, and a block matrix B with s row partitions and p column partitions are

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,s} \\ A_{1,1} & A_{1,2} & \cdots & A_{1,s} \\ \vdots & \vdots & \ddots & \vdots \\ A_{q,1} & A_{q,2} & \cdots & A_{q,s} \end{bmatrix}$$

and

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,p} \\ B_{1,1} & B_{1,2} & \cdots & B_{1,p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{s,1} & B_{s,2} & \cdots & B_{s,p} \end{bmatrix},$$

respectively. The matrix product $C = AB$ can be formed blockwise, yielding a $m \times n$ matrix C with q row partitions and r column partitions, where

$$C_{i,j} = \sum_{k=1}^{p} A_{i,k} \cdot B_{k,j},$$

and $1 \leq i \leq q$ and $1 \leq j \leq r$. Block product can only be calculated if blocks of matrices A and B are compatible, i.e., when the number of columns of the block $A_{i,k}$ equals to the number of rows of the block $B_{k,j}$, for each $1 \leq i \leq q$, $1 \leq j \leq r$, and $1 \leq k \leq p$. In what follows we consider $m = n = l$ and $p = q = r$ as well as that p divides n: such blocks are always compatible.

III. Experiments

In this section we discuss several dataflow algorithms for various problems, all of which use matrices and/or vectors for their storage of data. Our main focus is on the experimental evaluation of different dataflow techniques and data choreographies.

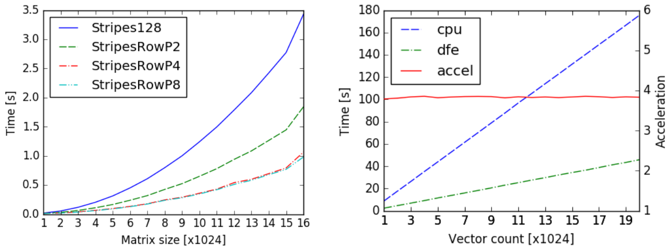
For the experiments we used Maxeler's Vectis MAX3424A PCI-express extension card, which contains dataflow unit based on Xilinx Virtex 6 SXT475 field-programmable gate array. The control-flow part of the computer contained Intel i7-6700K processor with 8 MB and 64 GB of cache and main memory, respectively.

A. Mutliplication of Matrices and Vectors

First we discuss the problem of multiplying a matrix with a vector from the dataflow perspective. Let $A = [a_{i,j}]$ be a $m \times n$ matrix and $B = [b_i]$ a vector of size n. The result of multiplying the matrix A with the vector B is a vector $C = [c_i]$ of dimension m, where

$$c_i = \sum_{j=1}^{n} a_{i,j} c_j .$$

Either rowwise, columnwise, or stripewise processing of the matrix may be used when considering dataflow algorithm for the problem. In Fig. 3 a) we give plots of running time versus matrix size for several variants of the stripe-based processing using rowwise replication of computation. Here the used stripe width is $s = 128$ while the number of pipes is 1 (no replication, denoted with Stripe128), 2 (StripeRowP2), 4 (StripeRowP4), and 8 (StripeRowP8). Observe that, doubling the number of pipes effects in halving the running-time performance, except for the 8 (or more) pipes, where due to the maximized throughput no improvement is observable.



a) Single vector, stripes

b) Multi-vector multiplication

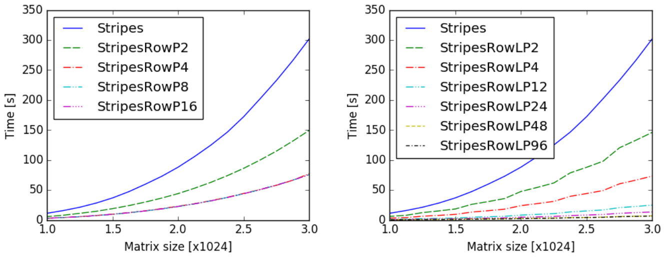
Fig. 3. Running-time performance for matrix-vector multiplication

In the comparison we excluded the control-flow algorithm, which outperformed all dataflow variants. Nevertheless, this is to be expected as there are not enough operations per a streamed element. Despite this, a use case to consider dataflow approach can be found in the setting where many vectors are to be multiplied with a given matrix. Here, the matrix is stored in the large memory of the dataflow engine, which has significantly higher streaming throughput than PCIe bus; hence, the results are now in favor of the dataflow architecture. See Fig. 3 b) for a graphical comparison: the left-hand side y-axis gives a running time in seconds while the right-hand side y-axis gives the acceleration factor obtained (about 4). In the experiment we run both algorithms (dataflow and control-flow), where the number of vectors varies from 1024 to 20480 in steps of 1024.

B. Matrix Multiplication

Now we present the results of experimental evaluation of the matrix multiplication problems using the above presented techniques for data choreography. In our experiments we vary the matrix dimension $n \times n$ from $n = 1024$ to $n = 3072$ in steps of 128 for slower algorithms as well as from $n = 1024$ to $n = 10240$ in steps of 512 for faster ones.

Here, we discuss stripe-based matrix access with rowwise pipe replication of computation, which is employed for streaming the left matrix of the multiplication problem while the right matrix is streamed in the columnwise fashion. Hence, we basically obtain n matrix-vector multiplication problems which are computed by the dataflow engine.



a) PCIe

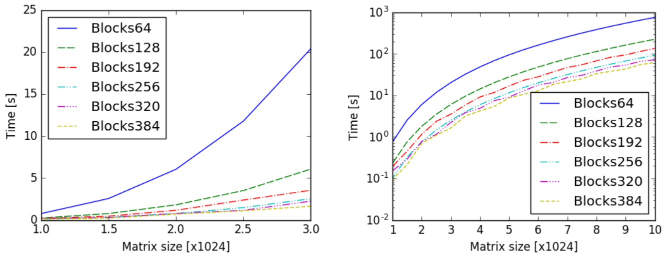
b) Large memory

Fig. 4. Running-time comparison of stripped matrix-access depending to the number of pipes

In Fig. 4 a) we show the results of the experiment where the data is streamed from the main memory to the dataflow engine using the PCIe bus, and in Fig. 4 b) where instead of the main memory the dataflow large memory is used. In the latter case, the matrix is only transfered once from the main memory to the large memory.

We can observe that the number of pipes has a desired effect on the performance. However, in the former technique up to 16 pipes are used, but 4 or more pipes do not exhibit any significant improvement for the throughput of the PCIe bus is already maximized. In the latter up to 48 pipes are used, each increase in pipe count causing an observable improvement in the performance. Here, the 48 pipes case is the last to give a performance improvement.

Now let us focus on the block-based matrix multiplication, which also has the greatest potential. As can be observed in

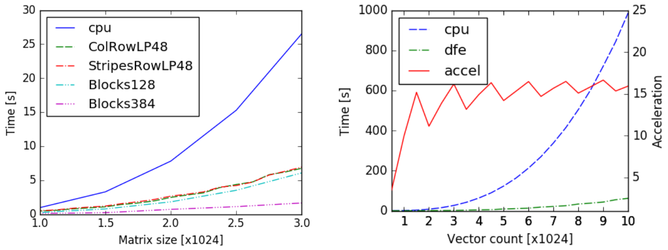


a) Small b) Large (log scale)

Fig. 5. Running-time comparison of block-based matrix multiplication depending on the block size

Fig. 5 a) these group of algorithms were much better. To show the scalability and practical usefulness of the algorithm when used with larger matrices we also show a graph of the running-time performance up to matrices of size 10240. Obviously, the larger the block size the better the performance achieved. See Fig. 5 b) for the corresponding graphical comparison, where the ordinate axis uses logarithmic scale.

In Fig. 6 a) we give a comparison of a representative data-choreography techniques. Namely, we include the best from each group: rowwise replication with columnwise (ColRowLP48) and stripe (StripesRowLP48) access, block-multiplication with block sizes of 128 (Blocks128) and 384 (Blocks384) as well as the control-flow implementation of the algorithm to give a better overview on the comparison. Notice that, the control-flow implementation is not highly optimized; however, we employed the classical technique of transposing the second matrix before multiplication, in order to get better performance of the cache memory due to the decrease of cache misses.



a) Various b) Acceleration

Fig. 6. Running-time comparison of various data choreographies

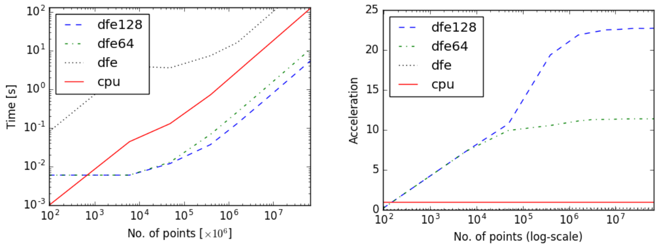
All the (selected) dataflow algorithms are better in running-time performance than control-flow algorithm. The columnwise and stripped-based techniques perform very similarly with the same number of pipes while block multiplication with the block size at least 128 × 128 outperforms all the other algorithms. Observe also that, one of the slowest block-based algorithm Block128 runs for about 6 seconds while the fastest stripped-based algorithms, StripesRowLP96, runs for about 7 seconds, when the matrix size is 3072 × 3072.

To finish with the comparison of the running time performance, let us have a look on the potential acceleration, which can be obtained with the dataflow-based algorithm. See Fig. 6 b) for the plot of the running time (left-hand side axis) and acceleration (right-hand side axis): the control-flow algorithm (cpu) and the best dataflow algorithm (dfe) are compared. Observe that, the acceleration of the dataflow over control-flow algorithm is about 15-fold.

C. Polynomial Evaluation

Let us now focus on the multi-point polynomial evaluation problem. Here the input is a polynomial (i.e., a stream of coefficients if the polynomial is dense, or a stream of exponents and coefficients if the polynomial is sparse), and a stream of points in which the polynomials is to be evaluated. Control-flow solutions are usually based on the well-known Horner algorithm [15]. Dataflow algorithms explore the data choreography ideas similar to as presented in Section II. Here we leave the technical details out, and present only the results of experiments.

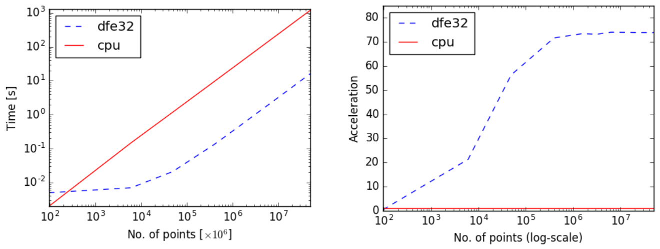
See Fig. 7 – a) for the running-time plots, and 7 b) for the acceleration plots – for the results of evaluation of dense polynomials with 1024 coefficients. Three dataflow algorithms are included: no pipes (dfe), 64 pipes (dfe64), and 128 pipes (dfe128). Observe, that without pipe-based replication the dataflow would be slower than control-flow algorithm. Nevertheless, using 64 or 128 pipes gives about 11- or 22-fold acceleration, respectively.



a) Running-time (log-log scale) b) Acceleration (log-x scale)

Fig. 7. Running-time of dense polynomial evaluation

Now we switch to sparse polynomials. See Fig. 8 a) and b) for running-time and acceleration plots, respectively. Again, the dataflow approach with 32 pipes outperformed the control-flow. In particular, the acceleration is about 70-fold.



a) Running-time (log-log scale) b) Acceleration (log-x scale)

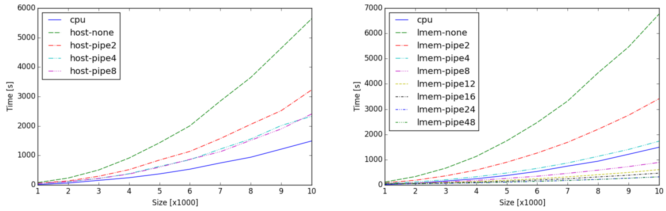
Fig. 8. Running-time of sparse polynomial evaluation

D. Simplex Pivoting

In this subsection we focus on a different problem, which is the main part of a classical simplex algorithm for finding the optimal solutions of linear programs. Here, a linear program in a canonical form is to optimize cx considering the constraints $Ax \leq b$, where c is an n-dimensional vector of the coefficients of a linear objective function, and A is the matrix of the coefficients of linear constraints; similarly b is a vector representing the coefficients of the right-hand side of the constraints.

In order to solve such linear system, the simplex algorithm repeatedly transforms the matrix until the optimal solution is found. Such transformations are based on the pivoting operation, which recalculates the matrix based on the selected (pivot) row and column.

We have implemented several versions of the pivoting operations: both variants of streaming from the main memory and from the large memory as well as variations on the number of pipes [9]. The results of the experiments are shown in Fig. 9 a) for streaming from the main memory via PCIe bus, and Fig. 9 b) for streaming from the large memory of the dataflow engine.



a) PCIe

b) Large memory

Fig. 9. Running-time comparison of various implementations of simplex pivoting

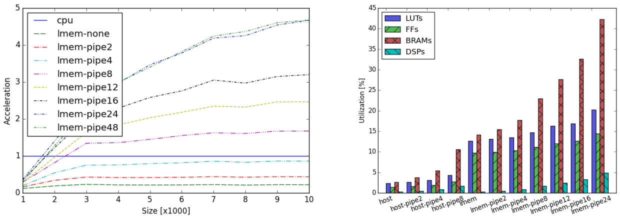
While the large-memory variant is able to achieve much greater accelerations (by using up to 24 pipes) it is also more complex to implement (consists of several kernels since the selection of the pivot column is done by the dataflow engine).

Finally, see Fig. 10 a) for accelerations (over the control-flow algorithm) of the dataflow algorithms streaming from the large memory. Observer, that 4 pipes are inadequate for the dataflow to outperform the control-flow. Hence, 8 or more are suggested while 24 already hit the transfer rate bottleneck of the large memory.

In Fig. 10 b) we also give resource consumption comparison for all of the implementations. Observe that large memory variants require much more resources than the ones where the data is streamed from the host. Additionally, increasing the pipes also increases the resources needed.

IV. Conclusion

For this paper we made a plethora of experiments with various dataflow algorithms and their variations. Our goal was to determine the techniques which work well with particular algorithms and problems. In the paper we presented a selected set of results, which we considered interesting.



a) Acceleration

b) Resource consumption

Fig. 10. Acceleration and resource consumption of various implementations of simplex pivoting

Acknowledgment

Our gratitude goes to prof. Veljko Milutinovć who introduced us to dataflow computing as well as Nemanja Trifunović for allowing us to use the dataflow computer and Ivan Milanković for helping us handling the computer. Most of the tests were performed by Matej Žniderič (matrix and vector multiplications) and Anže Sodja (polynomials).

References

[1] Jurij Šilc, Borut Robič, and Theo Ungerer. *Processor Architecture: From Dataflow to Superscalar and Beyond* Springer-Verlag Berlin Heidelberg, 1999.

[2] Ali R. Hurson and Veljko Milutinovi. *Dataflow Processing*, volume 96 of *Advances in Computers*. Elsevier, 2015.

[3] Maxeler Technologies Inc. Maximum performance computing. http://www.maxeler.com. Accessed: 2017-02-17.

[4] Anton Kos, Sao Tomai, Jakob Salom, and Veljko Milutinovi. New benchmarking methodology and programming model for big data processing. *International Journal of Distributed Sensor Networks*, 2015.

[5] Michael J. Flynn, Oskar Mencer, Veljko Milutinovi, Goran Rakoevi, Per Stenstrom, Roman Trobec, and Mateo Valero. Moving from petaflops to petadata. *Communications of the ACM*, 56(5):39–42, 2013.

[6] Nemanja Trifunovic, Veljko Milutinovic, Jakob Salom, and Anton Kos. Paradigm shift in big data supercomputing: Dataflow vs. controlflow. *Journal of Big Data*, 2(1):1–9, 2015.

[7] Vukašin Ranković, Anton Kos, and Veljko Milutinović. Bitonic merge sort implementation on the maxeler dataflow supercomputing system. *The IPSI BgD Transactions on Internet Research*, 9(2):5–10, 2013.

[8] Anton Kos, Rankovi, and Sao Vukain, Tomai. Sorting networks on maxeler dataflow supercomputing systems. *Advances in Computers*, 96:139–186, 2015.

[9] Uroš Čibej and Jurij Mihelič *Adaptation and Evaluation of the Simplex Algorithm for a Data-Flow Architecture*, in Advances in Computers, vol. 106, pp. 63–105, Elsevier, 2017.

[10] M. Müller-Hannemann and S. Schirra. *Algorithm engineering: bridging the gap between algorithm theory and practice*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010.

[11] Catherine C. McGeoch. *A guide to experimental algorithmics*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

[12] Jurij Mihelič and Uroš Čibej. Experimental algorithmics for the dataflow architecture: Guidelines and issues. *IPSI BgD Transactions on Advanced Research*, 13(1):1–8, 2017.

[13] V. Milutinović, J. Salom, N. Trifunović, and R. Giorgi. *Guide to DataFlow Supercomputing: Basic Concepts, Case Studies, and a Detailed Example*. Computer Communications and Networks. Springer International Publishing, 2015.

[14] Donald E. Knuth. *The Art of Computer Programming Volume 1: Fundamental Algorithms*, 3rd edition, section 2.2.6. Addison-Wesley: New York, 1997.

[15] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*, 2nd edition. McGraw-Hill Higher Education, 2001.

[16] PCI-SIG: Peripheral component interconnect special interest group. http://pcisig.com/, 2015. Accessed: 2017-02-17.

Model of the Telegraph Line

Gabriela Nečasová*, Petr Veigend*, Václav Šátek*† and Jiří Kunovský*
*Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 612 66
Brno, Czech Republic
(inecasova,iveigend,satek,kunovsky)@fit.vutbr.cz
†IT4Innovations, VŠB Technical University of Ostrava
17. listopadu 15/2172, 708 33
Ostrava-Poruba, Czech Republic
vaclav.satek@vsb.cz

***Abstract*—The paper deals with a model of the telegraph line, that consists of system of ordinary differential equations, rather than the partial differential telegraph equation. Computation is based on an original mathematical method. This method uses the Taylor series for solving ordinary differential equations in a non-traditional way.**

I. Introduction

The second order partial differential equations (PDEs) and especially the linear ones (elliptic, hyperbolic and parabolic) are very important in practical applications [6], [5]. One of these applications includes the telegraph equation, which is a PDE, that describes a telegraph line (a long wire that serves as a transmission medium for a signal). The PDE can describe the behaviour of the signal, however, this description does not contain any specific information about the conditions on the wire and it is very complicated. This paper presents a numerical model of the telegraph line, that consists only of ordinary differential equations (ODEs), it is relatively simple, easy to configure and the results match real output precisely. The system of ODEs is solved using the Modern Taylor Series Method (MTSM), which is introduced in the one of the sections of the paper. To show the suitability of the MTSM to solve this kind of problem and its advantages over the other commonly used methods (Runge-Kutta, Gear), the set of experiments is performed and the results are analysed.

II. Solution of Telegraph Equation

Voltage and current change along the telegraph line continuously in time and they can be expressed using equations

$$u = u(x,t), \tag{1}$$
$$i = i(x,t), \tag{2}$$

where x is a distance from the beginning of the line and t is the time. Voltage and current in the distance $x + \mathrm{d}x$ can be expressed using Taylor series with second and higher derivatives omitted

$$u(x+\mathrm{d}x) = u(x,t) + \frac{\partial u}{\partial x}\mathrm{d}x, \tag{3}$$
$$i(x+\mathrm{d}x) = i(x,t) + \frac{\partial i}{\partial x}\mathrm{d}x. \tag{4}$$

Basic Line Equations (5), (6) describe the change of voltage and current on the line.

$$-\frac{\partial u}{\partial x} = R_0 i + L_0\frac{\partial i}{\partial t} \tag{5}$$
$$-\frac{\partial i}{\partial x} = G_0 u + C_0\frac{\partial u}{\partial t} \tag{6}$$

Using (5) and (6) it is possible to construct a model of the segment. The entire line is then a series of infinite number of connected segments (Figure 1).

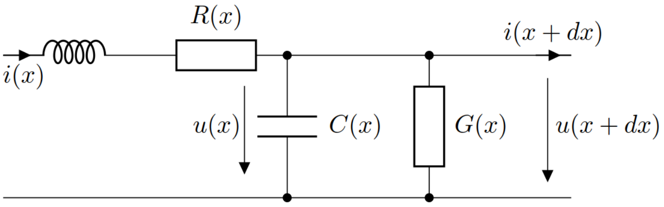


Figure 1. Modelling a segment of the wire – complex model

The model in the Figure 1 can be simplified by removing the resistance of the wire ($R(x)$ in the Figure 1) and the conductance between wires ($G(x)$ in the Figure 1). The simplified model is in the Figure 2.

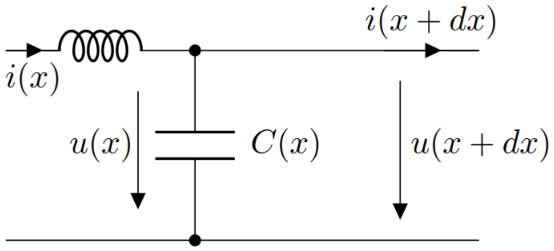


Figure 2. Modelling a segment of the wire – simplified model

Based on the simplified model, partial differential equations for voltage and current can be derived

$$L\cdot C\frac{\partial^2 u(x,t)}{\partial t^2} - \frac{\partial^2 u(x,t)}{\partial x^2} = 0, \tag{7}$$
$$L\cdot C\frac{\partial^2 i(x,t)}{\partial t^2} - \frac{\partial^2 i(x,t)}{\partial x^2} = 0. \tag{8}$$

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

The equations (7) and (8) can be solved analytically using for example the method of Separation of Variables, however the solution of large systems of PDEs is very complicated. Therefore, the simple model was created to model the telegraph equation. The model is described in Section IV.

III. Modern Taylor Series Method

The Modern Taylor Series Method (MTSM) is based on a recurrent calculation of the Taylor series terms for each time step [10]. Therefore the complicated calculation of higher order derivatives (much criticised in the literature) does not need to be performed but rather the value of each Taylor series term is numerically calculated. An important part of the method is an automatic integration order setting, i.e. using as many Taylor series terms as the defined accuracy requires. Thus it is usual that the computation uses different numbers of Taylor series terms for different time steps of constant length.

An automatic transformation of the original problem is a necessary part of the MTSM. The original system of differential equations is automatically transformed to a polynomial form, i.e. to a form suitable for easy calculation of the Taylor series terms using recurrent formulae.

The MTSM also has some properties very favourable for parallel processing [7]. Many calculations are independent, making it possible to perform the calculations independently using separate processors of parallel computing systems. Similar research with variable-step size and variable-order scheme is also described in [2] and [1], where simulations on a parallel computer are shown.

IV. Telegraph Equation Model

As stated in Section II, the model used in the experiments is the simplified model of the telegraph line (segment in Figure 2). By chaining these segments together, we can create a model of the line (Figure 3). Let us denote the number of segments of telegraph line used in the experiment as N.

The equations describing the model are below. For the first segment

$$\begin{aligned} u_0 &= \sin(\omega t), \\ u_{C1}^{'} &= \frac{1}{C_1}(i_1 - i_2), \\ i_1^{'} &= \frac{1}{L_1}(u_0 - u_{C1} - R_1 \cdot i_1), \end{aligned} \tag{9}$$

where u_0 is the harmonic input voltage of the system, u_{C1} is the voltage on the first capacitor and i_1 is the current that flows through the first inductor. Resistor R_1 represents input load of the transmission line. Equations for the following segments are very similar to one another. For the second segment

$$\begin{aligned} u_{C2}^{'} &= \frac{1}{C_2}(i_2 - i_3) \\ i_2^{'} &= \frac{1}{L_2}(u_{C1} - u_{C2}) \end{aligned} \tag{10}$$

for the next segments

$$\begin{aligned} u_{Ck}^{'} &= \frac{1}{C_k}(i_k - i_{k+1}), \\ i_k^{'} &= \frac{1}{L_k}(u_{Ck-1} - u_{Ck}), \end{aligned} \tag{11}$$

where $k \in \langle 3, N \rangle$. The last segment of the line ends with an output load, simulated by the resistor R_2.

$$i_{101} = \frac{1}{R_2} u_{C100} \tag{12}$$

Note that all differential equations have initial conditions equal to zero.

A. *Models used for experiments*

The experiments in the paper were performed using several models of the system (MATLAB and SPICE) presented in this section.

1) MATLAB: MATLAB is the language of scientific computing [12]. It contains the full suite of tools that can be used to solve various engineering and mathematical problems. Experiments in this paper use the ODE solvers, that are bundled with MATLAB [11]. The solvers used in the paper are based on the Runge-Kutta method (solvers ode23 and ode45) [3]. Solver ode45 is based on an explicit Runge-Kutta (4,5) formula, solver ode23 on an explicit Runge-Kutta (2,3) formula. Both are single-step solvers. Solver ode15s is used for stiff systems. It is a variable-step, variable-order (VSVO) solver based on the numerical differentiation formulas of orders 1 to 5. Solver ode15s is implemented as a multi-step solver.

MATLAB model is used for all experiments that use MATLAB solvers and newly implemented MTSM linear ODE solver. The model is represented by the linear system of ODEs in the form

$$\vec{x}' = \mathrm{A} \cdot \vec{x} + \vec{b}, \quad \vec{x} = \vec{0}, \tag{13}$$

where A is the sparse matrix, $\vec{x}$ is the vector of voltages and currents and $\vec{b}$ is the load vector that contains the input voltages. The block structure of matrix $\mathbf{A}$, vectors $\vec{x}$ and $\vec{b}$ follows

$$\mathrm{A} = \left(\begin{array}{c|c} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \hline \mathrm{A}_{21} & \mathrm{A}_{22} \end{array}\right), \vec{x} = \left(\begin{array}{c} u_{C1} \\ \vdots \\ u_{CN} \\ \hline i_1 \\ \vdots \\ i_N \end{array}\right), \vec{b} = \left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline \frac{u_0}{L_1} \\ \vdots \\ 0 \end{array}\right) \tag{14}$$

where A_{11}, A_{12}, A_{21} and A_{22} are individual block matrices

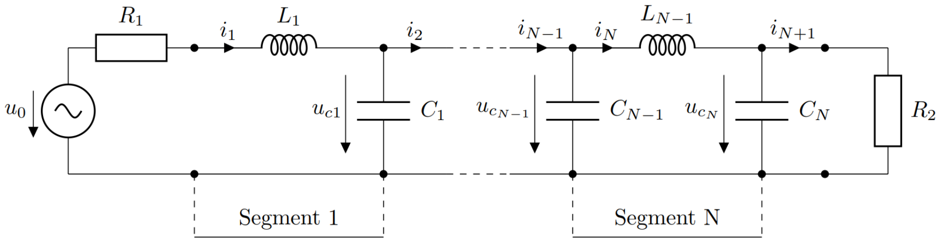


Figure 3. Model of the line – series of N segments

with size $N \times N$ (15).

$$A_{11} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{-1}{R_2 C_N} \end{pmatrix}$$

$$A_{12} = \begin{pmatrix} \frac{1}{C_1} & \frac{-1}{C_1} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{1}{C_2} & \frac{-1}{C_2} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \frac{1}{C_N} \end{pmatrix}$$

$$A_{21} = \begin{pmatrix} \frac{-1}{L_1} & 0 & 0 & \cdots & \cdots & 0 \\ \frac{1}{L_2} & \frac{-1}{L_2} & 0 & 0 & \cdots & \vdots \\ 0 & \frac{1}{L_3} & \frac{-1}{L_3} & 0 & \cdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \frac{1}{L_N} & \frac{-1}{L_N} \end{pmatrix}$$

$$A_{22} = \begin{pmatrix} \frac{-R_1}{L_1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{15}$$

This linear system of ODEs can be effectively solved in MATLAB using vectorization [13]. For our simulation experiments, the capacitances and inductances are the same, $C_1 = C_2 = \cdots = C_N$ and $L_1 = L_2 = \cdots = L_N$ (homogeneous telegraph line).

2) SPICE: Content of this section is mostly from [4]. SPICE is used for analogue circuit simulation because it can compute the full large signal behaviour of arbitrary circuits. SPICE uses three numerical methods for numerical integration. Newton integration is suitable to find the solution of circuits with non-linear elements. The sparse matrix method is used to save memory by storing only non-zero elements of the matrices. Implicit integration method is used to integrate the differential equations that describe the circuit.

Numerical integration of differential equations is necessary for analogue circuit simulation. SPICE uses second order integration. Most SPICE implementations follow Berkeley SPICE and provide two forms of second order implicit integration: Gear and trapezoidal (trap). Trap integration is both faster and more accurate than Gear, however trap integration can cause a numerical artefacts. These artefacts manifest like the oscillation around the precise solution in each time step. LTSpice was used for the experiments, it also provides Modified Trap method, which solves the problem with numerical artefacts.

The model for SPICE consists of components (models) for resistors, capacitors, inductors and input voltage (sine wave). The abbreviated version of the netlist (the description of the circuit for the model of 100 segments) is below (Code 1).

```
V1 1 0 SINE(0 1 1000000000.0)
R1 1 2 100.0
C1 2 0 1p
L1 2 3 10n
C2 3 0 1p
L2 3 4 10n
...
C99 100 0 1p
L99 100 101 10n
C100 101 0 1p
L100 101 102 10n
R100 102 0 100.0
```

Code 1. SPICE source code

V. Experiments

This section presents a set of experiments with the model from the section IV. The experiments are performed using the traditional (MATLAB ODE solvers and SPICE) and non-traditional approaches (new implementation of MTSM in MATLAB). The aim is to compare the efficiency of MTSM and other available methods and highlight the suitability of MTSM for solving these kinds of problems [8], [9]. All experiments were performed on 4 core CPU clocked at 3.2 GHz using MATLAB 2016b and LTSpice 7.

In practice, the telegraph line has to be adjusted for maximum efficiency. The line is adjusted when the condition stated by equation (16) is met.

$$R_1 = R_2 = \sqrt{\frac{L}{C}} \tag{16}$$

The experiments were performed on the adjusted line and on the non-adjusted line to highlight potential problems. The input voltage (u_0) has two possible configurations. The first configuration is the harmonic signal represented by sine wave $u_0 = \sin(\omega t)$, where $\omega = 10^9$ rad/s. The second one represents the impulse input and is defined by the following condition: $\text{if}(t < 1.1 \cdot 10^{-9}) \quad u = u_0 \quad \text{else} \quad u = 0$.

Simulation time for all experiments (if not specified differently) is $t \in \langle 0, t_{max} \rangle$ where $t_{max} = 2 \cdot 10^{-8}$.

The first experiment (Figure 4) shows the behaviour of the adjusted telegraph line for sine input (Figure 4) and impulse input (Figure 5). The voltage on the first capacitor (u_{C1}, solid line) is just slightly shifted version of the input signal with smaller amplitude (in this case, the amplitude is halved). The output signal (u_{C100}, dashed line) copies the signal on the first capacitor and shifts it further in time.

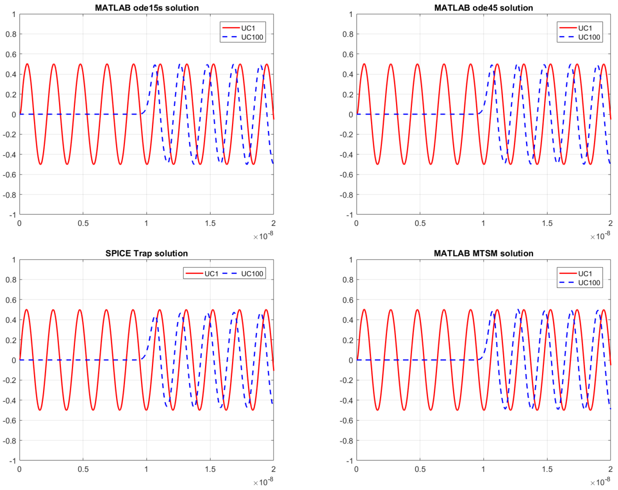


Figure 4. Adjusted line, sine input, output is just delayed

If the input of the system changes to the impulse (Figure 5), the voltage on the first capacitor u_{C1} is again just slightly shifted and its amplitude is decreased. The impulse then propagates through the line and is delayed on the last capacitor u_{C100}. The oscillations are also visible.

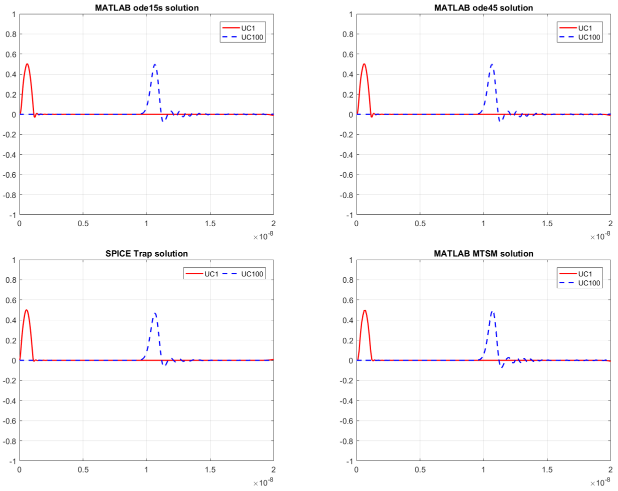


Figure 5. Adjusted line, impulse input, output is just delayed

The second experiment tests the behaviour for the non-adjusted telegraph line (condition (16) does not hold). For the sine input, the voltage on the first capacitor u_{C1} is again just very slightly shifted and halved. The voltage on the last capacitor u_{C100} is however delayed and amplified, which can cause damage to the connected equipment (Figure 6).

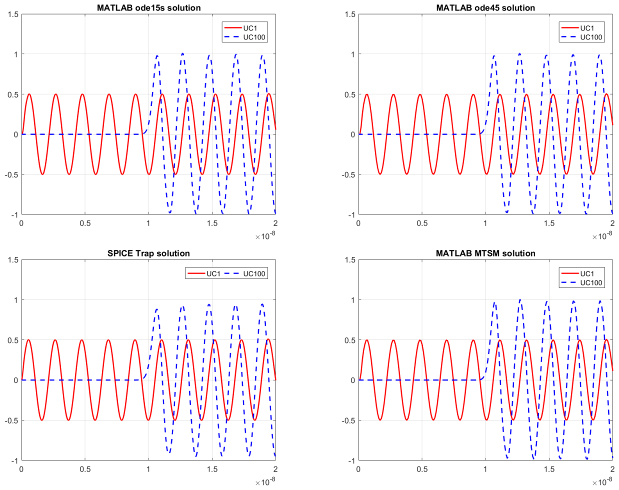


Figure 6. Non-adjusted line, sine input, output is delayed and amplified

For the impulse input, the output signal is delayed and amplified (Figure 7). . The input signal bounces back and appears again on the first capacitor. This behaviour can be used to detect, where the line is cut (the output spike, u_{C100}). The simulation time is doubled ($t_{max} = 4 \cdot 10^{-8}$).

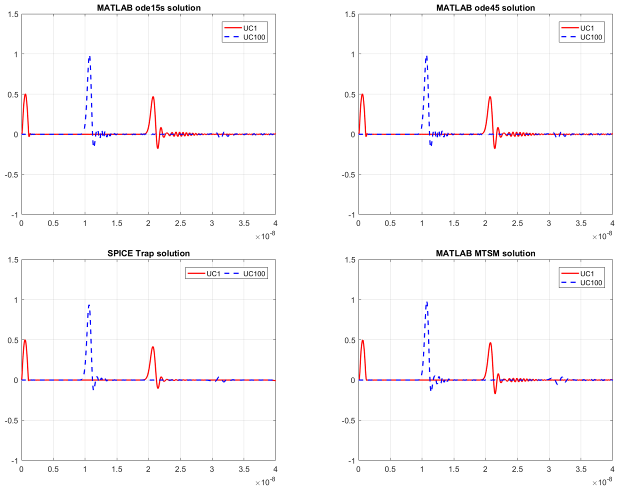


Figure 7. Non-adjusted line, impulse input, output is delayed, amplified and bounced back

A. Summary of conducted experiments

The following graph in the Figure 8 shows the number of integration steps needed by the solvers to solve the problem in the simulation time $t \in \langle 0, t_{max} \rangle$. It is obvious, that the MTSM needs the least amount of integration steps with the same or higher accuracy than the other methods.

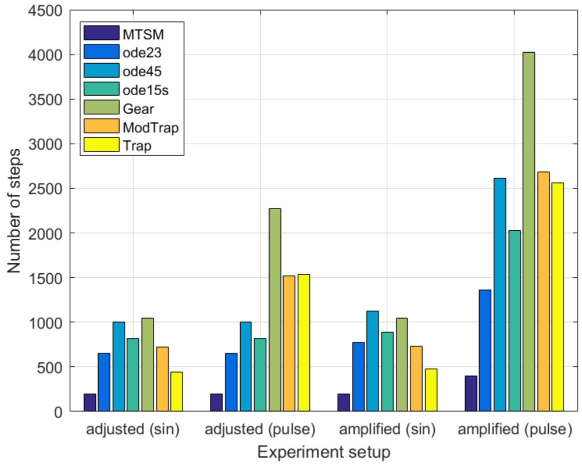


Figure 8. Comparison of number of steps, all methods

The following table shows the computation time for experiments in seconds (arithmetic average for 100 runs).

Experiment / Solver	ode15s [s]	ode45 [s]	MTSM [s]
adjusted telegraph line, sine input	$3.135066 \cdot 10^{-1}$	$3.234379 \cdot 10^{-2}$	$4.209659 \cdot 10^{-2}$
adjusted telegraph line, pulse input	$3.350448 \cdot 10^{-1}$	$3.206530 \cdot 10^{-2}$	$4.144095 \cdot 10^{-2}$
non–adjusted telegraph line, sine input	$3.696707 \cdot 10^{1}$	$3.714358 \cdot 10^{-2}$	$4.153812 \cdot 10^{-2}$
non–adjusted telegraph line, pulse input	$9.316806 \cdot 10^{-1}$	$9.047084 \cdot 10^{-2}$	$8.825860 \cdot 10^{-2}$

Table I. COMPUTATION TIME FOR EXPERIMENTS IN SECONDS

MTSM is implemented as variable-step size and variable-order method. For the experiments, the average order for the MTSM was automatically set to 16.

VI. CONCLUSION

This article deals with the numerical analysis of the second order partial differential equation – telegraph equation. This equation is replaced by the segment model and described by the system of ordinary differential equations (ODEs). The system has been solved numerically using MTSM and other methods for comparison.

As an example, the model with one hundred segments was presented. The experiments showed the behaviour of the system for the specific combination of parameters R_1 and R_2. For some combinations of parameters, bounces on the line could be observed. The experiments are summarised and clearly show the suitability of the MTSM for this kind of problems.

Due to the fact that the telegraph equation is the special case of the wave equation, it might be possible to slightly change the presented model to represent much wider class of problems, which is an interesting topic for future research. Due to the easy scalability of the problem, different approaches to parallelization can also be researched.

VII. ACKNOWLEDGEMENT

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602". The paper also includes the solution results of the internal BUT project FIT-S-17-4014 and AKTION project Aktion-76p11.

REFERENCES

[1] R. Barrio. Performance of the Taylor series method for ODEs/DAEs. In *Applied Mathematics and Computation*, volume 163, pages 525–545, 2005. ISSN 00963003.

[2] R. Barrio, F. Blesa, and M. Lara. VSVO Formulation of the Taylor Method for the Numerical Solution of ODEs. In *Computers and Mathematics with Applications*, volume 50, pages 93–111, 2005.

[3] John C Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2008.

[4] Mike Engelhardt. Spice differentiation. *ELECTRONICS WORLD*, 121(1946):16–21, 2015.

[5] L. C. Evans. *Partial differential equations; 2nd ed.* Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2010. ISBN 978–0821849743.

[6] S. J. Farlow. *Partial differential equations for scientists and engineers*. Courier Corporation, 2012. ISBN 860-1234581253.

[7] Filip Kocina, Jiří Kunovský, Gabriela Nečasová, Václav Šátek, and Petr Veigend. Parallel solution of higher order differential equations. In *Proceedings of the 2016 International Conference on High Performance Computing & Simulation (HPCS 2016)*, pages 302–309. Institute of Electrical and Electronics Engineers, 2016.

[8] J. Kunovský, V. Šátek, and A. Szöllös. Telegraph equation and corresponding wave forms. In *Proceedings of the 12th International Scientific Conference Electric Power Engineering 2011, EPE 2011*, pages 561–564, 2011.

[9] J. Kunovský, V. Šátek, D. Topolánek, and V. Vopěnka. Telegraph equation and its application in medium voltage line. In *Proceedings of the 13th International Scientific Conference Electric Power Engineering 2012, EPE 2012*, volume 1, pages 175–180, 2012.

[10] Jiří Kunovský. *Modern Taylor Series Method*. Habilitation thesis, FEKT VUT Brno, 1995.

[11] MATHWORKS. Choose and ode solver – matlab & simulink. https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html.

[12] MATHWORKS. Matlab - the language of technical computing. http://www.mathworks.com/products/matlab/.

[13] Václav Šátek, Filip Kocina, Jiří Kunovský, and Alexander Schirrer. Taylor series based solution of linear ode systems and matlab solvers comparison. In *MATHMOD VIENNA 2015 - 8th Vienna Conference on Mathematical Modelling*, ARGESIM REPORT No. 44, pages 693–694. ARGE Simulation News, 2015.

The simulation of paintings with different aesthetic variables Temperature and Harmony

Marie Nedvedova
University of Pardubice
Faculty of electrical engineering and informatics
Studentska 95, Pardubice, 532 10, Czech republic
Email: st36806@student.upce.cz

Jaroslav Marek
University of Pardubice
Faculty of electrical engineering and informatics
Studentska 95, Pardubice, 532 10, Czech republic
Email: jaroslav.marek@upce.cz

***Abstract*—The aesthetic characteristics Temperature and Harmony were suggested for measuring of aesthetic impression image composed of simple patterns. These characteristics are influenced by the number of different types of elements contained in the image, and the number of different symmetries in their arrangement. Full details of all steps in these calculations are given in this paper. The primary concern of this reseach is to investigate the two-dimensional domain of characteristics Temperature and Harmony that depends on the number of different elements. With the use of random generation and evolutionary algorithms, thousands of images will be generated.**

I. Introduction

Australian mathematician Nikos Salingaros together with Professor Allen Klinger suggested a set of aesthetic variables: L (Life — interest rate), C (Complexity — measure of randomness), T (Temperature — diversity of elements), H (Harmony — orderliness), S (Entropy — non-orderliness).

The constraint for this method is the location of each element of the patterns in regular square fields. It is measured by two basic image properties: the number of different types of elements and the number of symmetries in their arrangement [1].

But they did not present the relationship between the number of colors and the set of characteristics. So we will try to deal with this task in our paper.

Klinger and Salingaros stated, that aesthetic impression of image was determined by the variables Temperature and Harmony. They proposed the hypothesis, that by using measurements L and C for patterns in the images, it is possible to determine the overall impression of the image. But we will not discuss the hypotheses they have proposed. This will be the subject of further research.

Other methods for measuring aesthetic variables exist. Some other symmetry measures are suggested in the paper [2]. Creation of paintings with symetric patterns are investigated in [3]. A brief history of aesthetic measurement is given, for example, in papers [4], [5].

A. Aesthetic variables Temperature and Harmony

Klinger and Salingaros defined two complementary composite measures L and C, where L is Life and C is Complexity. These values are calculated for the image as a whole, and consequently for its sub-elements.

For illustration, we will describe the calculation of these characteristics for a painting which is composed of 64 elements, placed into a regular 8×8 grid. Characteristics are then computed, at first, for an 8×8 square and then for its sub-elements composed of 4×4 and 2×2 squares. For a painting, which is composed of 36 elements placed into a regular 6×6 grid, characteristics are calculated firstly for the 6×6 square, and then also for its sub-elements composed of 3×3 and 2×2 squares. Everything indicates that the value of characteristics L determines how much viewers feel the image ”interesting”. This could help to quantify the visual connection of two-dimensional patterns with the viewer. Characteristics C determine a rate of randomness of the array.

Images are either immediately understandable or not, on the basis of how they are processed by our mind. It depends not only on the content of the image as a whole, but also on relationships between its substantive elements. Among the operations leading to an immediate understanding of the image, belongs the perception of interrelated (mutually related) elements and the number of its repetitions. The degree of symmetry is determined by comparing the associated elements and their mutual locations in the image. The greatest creations of humans — whether it's buildings, cities, artwork or artefacts — are not simple, nor coincidental, but show a high degree of organized complexity [1].

We can number the elements in the painting to make it easier to refer to specific processes of aesthetic characteristic computation. See Fig. 1. Throughout the paper in the figures, we will show the values of aesthetic characteristics so that the reader can compare the images.

Klinger and Salingaros described characteristics L and C with the help of variables T and H. Variable T (Temperature) expresses the diversity of elements. In our case, it is calculated as the sum of different colours in the image (or in its subblock) minus one.

Variable H (Harmony) measures the correlation of subblocks using selected symmetry. We can also refer to H as negative entropy, because the presence of symmetry is linked with the absence of visual fragmentation.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

With characteristics T and H formulas for calculation of L and C are:

$$\begin{aligned} L &= TH, && (1)\\ C &= T(H_{max} - H), && (2) \end{aligned}$$

where H_{max} is maximal H, which can be achieved in a given system.

Entropy S can be calculated as

$$S = H_{max} - H, \qquad (3)$$

and so

$$C = TS. \qquad (4)$$

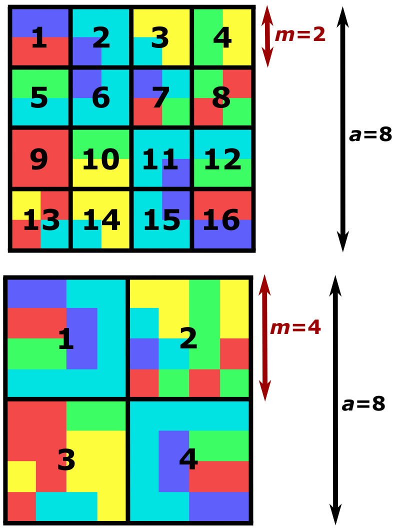
In practice, it is much easier to calculate H (given by the sum symmetry) than S, where it is necessary to determine the degree of disorder. Maximal symmetry H_{max} is a constant for each specific system, and for fixed T, composite measures L and C are different. The following formula applies:

$$C + L = TH_{max}. \qquad (5)$$

Why do we measure T, H at first and then we calculate L and C? Klinger and Salingaros believe that the characteristics T and H, which can be measured from an image, are not directly perceptible to an observer, but there are characteristics L and C, which create an overall impression of the image.

B. Example

Firstly we will compute values T and H, and then by putting them into formulas we will calculate L and C. For calculation T and H, we will have to split the sample image into smaller parts – subblocks (see Fig. 1).



$T = 2.792$, $H = 1.042$, $L = 2.908$, $C = 22.217$, colors = 5

Fig. 1. Labels of Elements, and Sample of Structures. Source: own (drawing in JAVA).

We calculate the partial $T(m \times m)$ for the image divided into sixteen subblocks with side 2×2, into four subblocks with side 4×4 and then for a whole image (block with side 8×8). Resulting T will be calculated from these partial $T(m \times m)$.

The general formula for calculation of partial $T(m \times m)$ can be written as:

$$T(m \times m) = \frac{1}{n_m} \sum_{i=1}^{n_m} T_i(m \times m), \qquad (6)$$

where m is length of the subblock side and n is the number of subblocks in the image. n can be also expressed as:

$$n = a^2/m^2, \qquad (7)$$

where a is the length of the picture and m is the length of the subblock side. T takes the value of the sum of different colours in each subblocks reduced by one.

Image divided into blocks 2×2

Particular $T(2\times2)$ for the image divided into subblock with sides 2×2 are calculated according to the formula:

$$T(2 \times 2) = \frac{1}{n_2} \sum_{i=1}^{n_2} T_i(2 \times 2). \qquad (8)$$

For Fig. 1 variable a is equal to 8 and variable m is equal to 2. We determine n_2 according to the formula (7):

$$n_2 = \frac{a^2}{m^2} = \frac{8^2}{2^2} = 16.$$

After substituting into the formula (8) we get:

$$\begin{aligned} &T(2 \times 2) = \frac{1}{16} \sum_{i=1}^{16} T_i(2 \times 2) = \frac{1}{16} (T_1(2 \times 2) + \ldots + \\ &+T_{16}(2 \times 2)) = \frac{1}{16} (1 + 1 + 1 + 1 + 1 + 1 + 3 + 1 + \\ &0 + 1 + 1 + 1 + 2 + 1 + 1 + 1) = \frac{18}{16} = 1.125. \end{aligned}$$

Image divided into blocks 4×4

Particular $T(4\times4)$ for the image divided into subblock with sides 4×4 are computed according formula:

$$T(4 \times 4) = \frac{1}{n_4} \sum_{i=1}^{n_4} T_i(4 \times 4). \qquad (9)$$

Firstly, we determine n_4 according to the formula (7):

$$n_4 = \frac{a^2}{m^2} = \frac{8^2}{4^2} = 4.$$

After substituting into the formula (9) we get:

$$\begin{aligned} &T(4 \times 4) = \frac{1}{4} \sum_{i=1}^{4} T_i(4 \times 4) = \frac{1}{4} \cdot (T_1(4 \times 4) + \ldots + \\ &+T_4(4 \times 4)) = \frac{1}{4} \cdot (3 + 4 + 3 + 3) = 13/4 = 3.25. \end{aligned}$$

Particular $T(8 \times 8)$ for the block with side 8×8 is computed according formula:

$$T(8 \times 8) = \frac{1}{n_8} \sum_{i=1}^{n_8} T_i(8 \times 8). \qquad (10)$$

We determine n_8 according to the formula (7):

$$n_8 = \frac{a^2}{m^2} = \frac{8^2}{8^2} = 1.$$

After substitution into the formula (10) we get:

$$T(8 \times 8) \quad = \quad \frac{1}{1} \sum_{i=1}^{1} T_i(8 \times 8) = T_1(8 \times 8) = 4.$$

Then we calculate T by the following formula:

$$T \quad = \quad \frac{1}{|D|} \sum_{m \in D} T(m \times m), \qquad (11)$$

where $|D|$ is number of partial $T(m \times m)$, which we have calculated for the image in the previous steps.

In our case we will get:

$$\begin{aligned} T \quad &= \quad \frac{1}{3}(T(2 \times 2) + T(4 \times 4) + T(8 \times 8)) = \\ &= \quad \frac{1}{3}(1.375 + 3.25 + 4) = 2.792. \end{aligned}$$

Calculation of H will be more demanding. Even here in the same way, we divide the image into the subblocks. Variable H measures the presence of symmetry.

We investigate nine kinds of symmetry. Let use labels h_1 – h_9, where h_1 – h_6 within one subblock represent:

- h_1 — reflective symetry along the axis x,
- h_2 — reflective symetry along the axis y,
- h_3 — reflective symetry along the axis of the first and third quadrant,
- h_4 — reflective symetry along the axis of the second and four quadrant,
- h_5 — rotational symetry with angle 90°
- h_6 — rotational symetry with angle 180°.

The other three cases are calculated relative to the other subblocks of the same dimension (in our case it only makes sense to calculate only within 2×2 and 4×4 subblocks):

- h_7 — the identity of the subblock to another subblock of the same size (translational consensus),
- h_8 — translational consensus with another subblock after reflection along the axis x or axis y,
- h_9 — translational consensus with another subblock after rotation of 90°, -90° or 180°.

Symetries h_1 — h_9 may be present or may not be present. Therefore, the value h may only take values 0 (symmetry is not present), or 1 (symmetry is present).

The resulting H value is determined similarly to the T value by calculation of partial $H(m \times m)$. Firstly, the characteristic $H(2 \times 2)$ of the image, that is divided into sixteen subblocks with side 2×2, is calculated. Then the characteristic $H(4 \times 4)$ of the image, that is divided into four subblocks with side 4×4, is determined. Finally, the characteristic $H(8 \times 8)$ of the whole image (blog with side 8×8) is computed. The resulting H will be calculated from these partial $H(i \times i)$, $i = 2, 4, 8$.

The general formula for calculation of partial $H(m \times m)$ can be written as:

$$H(m \times m) \quad = \quad \frac{1}{n_m} \sum_{i=1}^{n_m} H_i(m \times m), \qquad (12)$$

where m is length of the subblock side and n is the number of subblocks in the image. n_m can also be expressed as $\frac{a^2}{m^2}$, where a is the length of the picture and m the length of the side subblock. $H_i(m \times m)$ is the total sum of symmetries h_1 – h_9 measured from the subblock i.

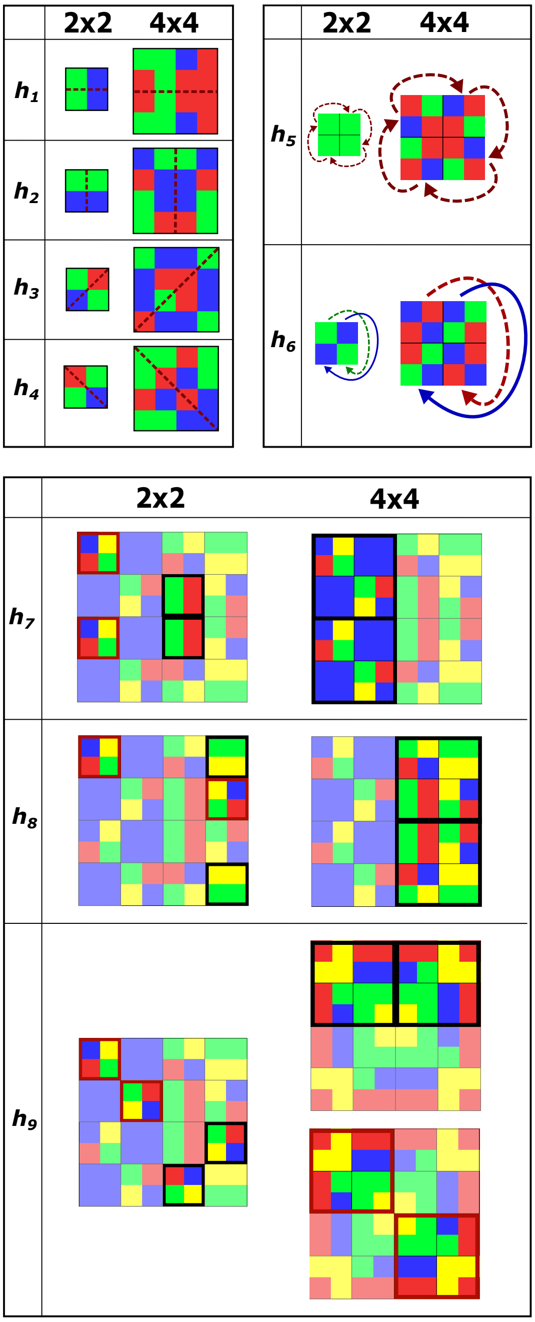


Fig. 2. Symmetries h_1 – h_9.

We will determine $H(2 \times 2)$ for the image divided into subblocks with the side 2×2:

$$H(2 \times 2) \;=\; \frac{1}{16}\left(H_1(2 \times 2) + \ldots + H_{16}(2 \times 2)\right), \tag{13}$$

where

$$\begin{aligned}
H_1(2 \times 2) &= h_1(1)(2 \times 2) + \ldots + h_9(1)(2 \times 2),\\
H_2(2 \times 2) &= h_1(2)(2 \times 2) + \ldots + h_9(2)(2 \times 2),\\
&\ldots\\
H_{(16)}(2 \times 2) &= h_1(16)(2 \times 2) + \cdots + h_9(16)(2 \times 2).
\end{aligned}$$

For better understanding, first we find $H_1(2{\times}2)$ — $H_{16}(2{\times}2)$:

$$\begin{aligned}
H_1(2 \times 2) &= 0+1+0+0+0+0+0+1+1 = 3,\\
H_2(2 \times 2) &= 0+0+1+0+0+0+0+1+1 = 3,\\
H_3(2 \times 2) &= 0+0+1+0+0+0+1+0+0 = 2,\\
H_4(2 \times 2) &= 1+0+0+0+0+0+0+0+1 = 2,\\
H_5(2 \times 2) &= 0+1+0+0+0+0+0+1+1 = 3,\\
H_6(2 \times 2) &= 0+0+0+1+0+0+0+1+1 = 3,\\
H_7(2 \times 2) &= 0+0+0+0+0+0+0+0+0 = 0,\\
H_8(2 \times 2) &= 0+0+1+1+0+1+0+0+0 = 3,\\
H_9(2 \times 2) &= 1+1+1+1+1+1+0+0+0 = 6,\\
H_{10}(2 \times 2) &= 0+1+0+0+0+0+0+0+1 = 2,\\
H_{11}(2 \times 2) &= 0+0+0+1+0+0+0+1+1 = 3,\\
H_{12}(2 \times 2) &= 0+1+0+0+0+0+0+1+1 = 3,\\
H_{13}(2 \times 2) &= 0+0+0+1+0+0+0+0+0 = 1,\\
H_{14}(2 \times 2) &= 0+0+1+0+0+0+1+0+0 = 2,\\
H_{15}(2 \times 2) &= 0+0+1+0+0+0+0+1+1 = 3,\\
H_{16}(2 \times 2) &= 0+1+0+0+0+0+0+1+1 = 3.
\end{aligned}$$

Now we can substitute into the formula (13) for calculating $H(2 \times 2)$:

$$\begin{aligned}
&H(2 \times 2) = \frac{1}{16} \cdot (H_1(2 \times 2) + \ldots + H_{16}(2 \times 2))\\
&= \frac{1}{16} \cdot (3+3+2+2+3+3+0+3+\\
&6+2+3+3+1+2+3+3) = \frac{1}{16} \cdot 42 = 2.625.
\end{aligned}$$

Similarly we calculate $H(4 \times 4)$:

$$H(4 \times 4) \;=\; \frac{1}{n_4}\sum_{i=1}^{n_4} H_i(4 \times 4). \tag{14}$$

For image divided into subblocks with side 4×4 we calculate H as follows:

$$H(4 \times 4) \;=\; \frac{1}{4} \cdot \left(H_{(1)}(4 \times 4) + \ldots + H_{(4)}(4 \times 4)\right), \tag{15}$$

where

$$\begin{aligned}
H_1(4 \times 4) &= h_1(1)(4 \times 4) + \ldots + h_9(1)(4 \times 4),\\
&\ldots\\
H_4(4 \times 4) &= h_1(4)(4 \times 4) + \ldots + h_9(4)(4 \times 4).
\end{aligned}$$

We will calculate $H_1(4 \times 4)$ – $H_4(4 \times 4)$:

$$\begin{aligned}
H_1(4 \times 4) &= 0+0+0+0+0+0+0+0+1 = 1,\\
H_2(4 \times 4) &= 0+0+0+0+0+0+0+0+0 = 0,\\
H_3(4 \times 4) &= 0+0+0+0+0+0+0+0+0 = 0,\\
H_4(4 \times 4) &= 0+0+0+0+0+0+0+0+1 = 1.
\end{aligned}$$

Now we can substitute into the formula (14) for calculating $H(4 \times 4)$:

$$\begin{aligned}
H(4 \times 4) &= \frac{1}{4}\left(H_1(4 \times 4) + \ldots + H_4(4 \times 4)\right)\\
&= \frac{1}{4}(1+0+0+1) = \frac{1}{4} \cdot 2 = 0.5.
\end{aligned}$$

Now we determine $H(8 \times 8)$:

$$H(8 \times 8) \;=\; \frac{1}{n_8}\sum_{i=1}^{n_8} H_i(8 \times 8), \tag{16}$$

after substitution we receive:

$$H(8 \times 8) = \frac{1}{1} \cdot H_1(8 \times 8). \tag{17}$$

Symetries h_7–h_9 will definitely be zero, because there are no other 8×8 blocks, which could be compared with each other. We enumerate $H_1(8 \times 8)$:

$$\begin{aligned}
H_1(8 \times 8) &= h_1(1)(8 \times 8) + \ldots + h_9(1)(8 \times 8) =\\
&= 0+0+0+0+0+0+0+0+0.
\end{aligned}$$

After substitution into the formula (17):

$$H(8 \times 8) = \frac{1}{1} \cdot (0+0+0+0+0+0+0+0+0) = 1 \cdot 0 = 0.$$

Then we calculate resulting H by the following formula:

$$H = \frac{1}{|D|}\sum_{m=1}^{|D|} H(m,m), \tag{18}$$

where $|D|$ is number of particular $H(m,m)$, which we have calculated for the image in the previous steps. Then

$$\begin{aligned}
H &= \frac{1}{3} \cdot (H(2 \times 2) + H(4 \times 4) + H(8 \times 8))\\
&= \frac{1}{3} \cdot (2.625 + 0.5 + 0) = 1.042.
\end{aligned}$$

Calculation L a C we already performed by simple substitution into formulas:

$$\begin{aligned}
L &= T \cdot H = 2.792 \cdot 1.042 = 2.908,\\
C &= T \cdot (H_{max} - H),
\end{aligned}$$

where H_{max} is maximal value of H, which can be achieved in a given system. Maximal symetry H_{max} is constant for every specific system.

Then

$$C = 2.792 \cdot (9 - 1.042) = 22.217.$$

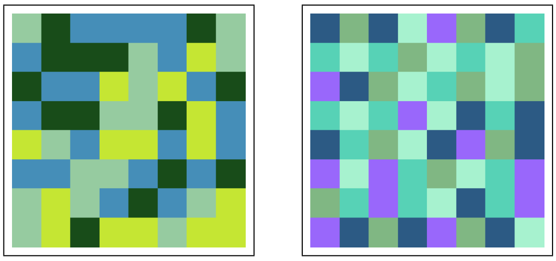
II. Dispute about the value range of characteristics L and C

For a closer examination of the Klinger-Salingaros hypothesis, it is necessary to specify the range of L and C characteristics.

To plot the value range of all L and C characteristics for a maximum four-color images, we would need to generate 4^{64} images, which is about $3{,}4 \cdot 10^{38}$ combinations. Even for plotting the L and C values for only two-color images, we would need to generate more than $1{,}8 \cdot 10^{19}$ images. Of course, this is almost impossible.

A. Generating images using a random number generator

During the analysis of the L and C characteristics of randomly generated images, it was found that the L and C values for a large part of the generated images are only displayed in a very limited area of the LC graph. In Fig. 5 is a graph of L and C values of randomly generated images. There were randomly generated 300 million images of two to four colors (100 million images for each number of colors). In the case of the group of generated images, the L and C values were often the same.



$T = 2.6250$, $H = 0.4167$
$L = 1.0938$, $C = 22.5313$
colors $= 4$

$T = 3.6667$, $H = 0.0000$
$L = 0.0000$, $C = 33.0000$
colors $= 5$

Fig. 3. Images generated using a random number generator.

In Fig. 5, we can see a graph of the L and C characteristics for 300 million randomly generated images, rendered in different colors according to the frequency of matches. The red points in the graph represent separate images, the yellow points represent the consensus of five images, the purple points represent the consensus of fifteen or more images (see the legend of the graph).

In Fig. 4, in detail we can see the matches of the generated images. A graph of relative occurrence frequencies is used to display obtained results.

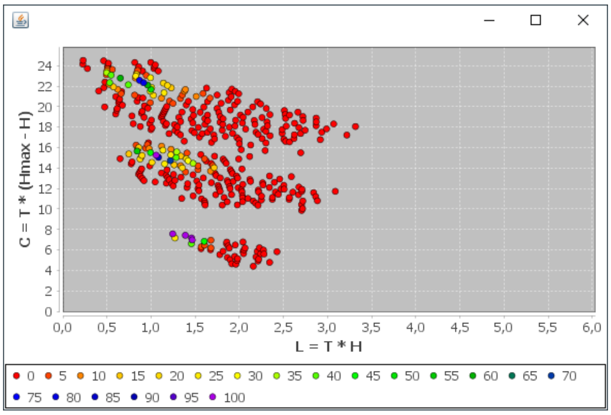


Fig. 4. Detail of match of characteristics L and C of randomly generated images (a graph of relative occurrence frequencies is used to display this).

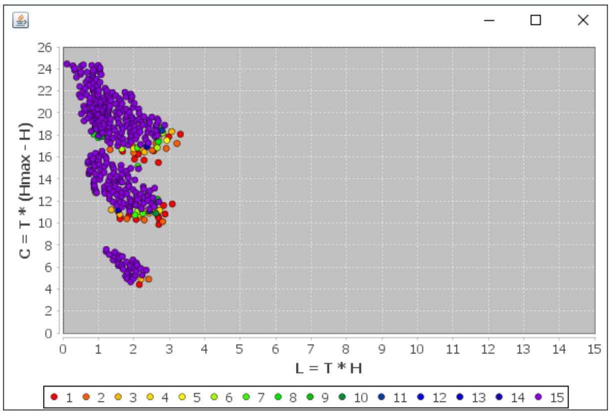


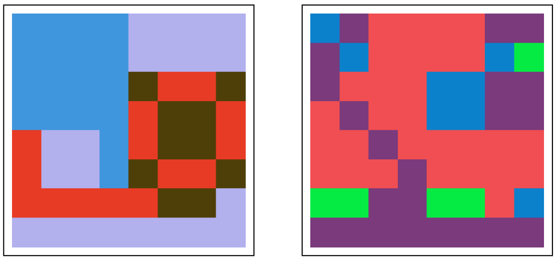
Fig. 5. Graph of characteristics L and C for 300 million randomly generated images (a graph of absolute occurrence frequencies).

Generating images, using only the random number generator, does not appear to be a suitable method to cover the entire field of values of characteristics L and C.

B. Generating images by pattern combining.

However, the range of L and C characteristics is considerably larger. With increasing symmetries, the L characteristic increases in certain circumstances. The C characteristic is most affected by the amount and spacing of each color in the image. However, L and C also have a certain dependence on each other, since both are calculated from the T and H characteristics by the formula (2). Therefore, it will be necessary to deliberately generate certain types of images to expand the value range.

Several ways of generating images have been used. First of all, pattern combining was done. Sets of 2×2 and 4×4 blocks have been created. Each block always contained a certain type of symmetry. The algorithm randomly selected the blocks from the prepared sets and then composed the resulting images.



$T = 1.7292$, $H = 2.4583$
$L = 4.2509$, $C = 11.3116$
colors $= 4$

$T = 2.0208$, $H = 2.0208$
$L = 4.0838$, $C = 14.1037$
colors $= 4$

Fig. 6. Images created by pattern combining.

Fig. 7 shows the L, C chart for 300 million randomly generated images and 100 million images created by pattern combining. A graph of absolute occurrence frequencies is used to display this.

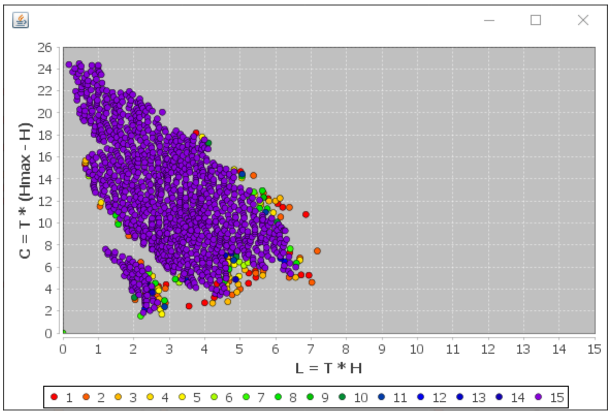


Fig. 7. Graph of L and C values of 300 million randomly generated images and 100 million images created by pattern combining (a graph of absolute occurrence frequencies).

C. Generating images using genetic algorithm

Next, the genetic algorithm was used to generate images.

The principle of the functioning of the genetic algorithm is mentioned in [6]. The genetic algorithm belongs to so-called evolutionary algorithms. It is a stochastic process that tries to find solutions of complex problems for which there is no usable exact algorithm. The principles of evolutionary biology are applied in this case.

Genetic algorithms have several characteristics:

1) They work with a whole range of possible solutions to the problem, instead of finding an individual solution,
2) they gradually improve the generated solutions by including new solutions; new solutions are obtained by combining (crossing) the original solutions,
3) Random changes (mutations) can occur in solutions,
4) unsatisfactory solutions are gradually eliminated.

A genetic algorithm in the beginning, has a certain population, where individuals of the population are various solutions to the problem. When crossing, the algorithm gradually generates new solutions to the problem. New individuals are born, who usually have a random portion of genes from one parent and the rest of the genes from the other parent. Each of the newly created individuals is calculated so. Fitness function expresses the quality of the solution represented by this individual. During the crossover, in the chromosome of an individual, a random change of a gene may occur (random mutation). This random mutation may or may not be favorable for further evolution of the species. The process of crossbreeding is constantly repeated, and generations with ever better genetic properties can emerge.

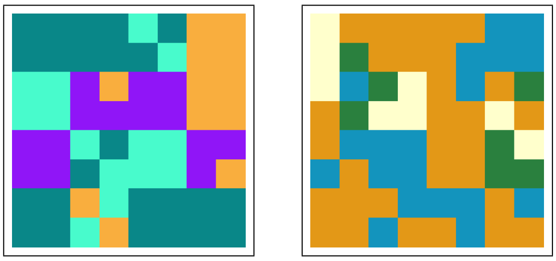
Specification of the used genetic algorithm:

The initial population was composed of a set of 1 000 images. The first third of the population consisted of images generated randomly. The second third of the population contained images created by pattern combining, while the 2×2 and 4×4 subblocks were placed at a random location in the image. The last third of the population contained images created by pattern combining. During the pattern combining process the subblocks were placed strictly at certain points so as to achieve even greater symmetries.

As a fitness function, a function computing the Euclidean distance to the target point $[L, C]$ was selected. The target point was chosen for each generated image randomly.

The population also changed during the calculation. Random mutations were possible. At the multipoint crossover, the two images were divided into three parts, some of which were swapped with one another. During the swap mutation, some points were randomly swapped within one image. The probabilities of occurrence of the mutations varied, and each generated image were different.

Evolutionary calculation was set to finish 2 000 generations. If a stabilization occurred, the calculation would be terminated after 500 iterations that did not improve the maximum fitness value.



$T = 2.1042$, $H = 2.6250$
$L = 5.5234$, $C = 13.4141$
colors $= 4$

$T = 2.1458$, $H = 1.5417$
$L = 3.3082$, $C = 16.0043$
colors $= 4$

Fig. 8. Images obtained with genetic algorithm.

Fig. 9 shows the L, C chart for 300 million randomly generated images, 100 million images created by pattern combining and 13 200 images created by using a genetic algorithm. Aa graph of absolute occurrence frequencies is used to display this.

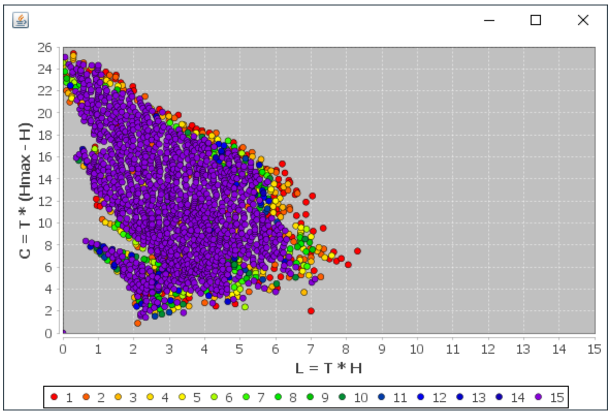
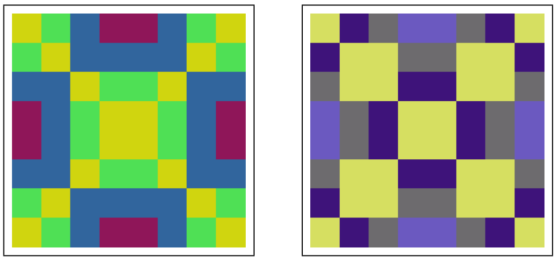


Fig. 9. Graph of L and C values of 300 million randomly generated images, 100 million images created by pattern combining and 13 200 images created by using genetic algorithm (a graph of absolute occurrence frequencies).

D. Purposeful generating images containing large amounts of symmetry.

A certain number of images especially at the top left and bottom left area of value range were created manually. For example, the highest values of characteristic C with the minimum L characteristic, were achieved by combining as many colors as possible in each 2×2 and 4×4 subblocks while trying to achieve zero or minimal symmetries. Gradually adding symmetries, the values of the L characteristic increased, and the upper boundary of the value range began to form. This always (verified for two to five color images) forms an angle of 135° with a horizontal axis.

As mentioned above, with increasing symmetries, the L characteristic increases in certain circumstances.



$T = 2.3333$, $H = 5.6667$
$L = 13.2222$, $C = 7.7778$
colors $= 4$

$T = 2.5000$, $H = 5.6667$
$L = 14.1667$, $C = 8.3333$
colors $= 4$

Fig. 10. Images generated to achieve high symmetries.

Therefore, the images with the highest values of L (the images at the boundary of the value range in the right area of the graph) will contain a large number of symmetries. This can be accomplished by focusing on generating images that will also contain h_1 and h_2 symmetry for block 8×8. If the 8×8 block is positive for the presence of symmetries h_1 and h_2 at the same time, then this 8×8 block must necessarily also have h_6 symmetry, and each of its subblocks 4×4 and 2×2 must be positive for the presence of symmetries h_8 and h_9. However, if a generated picture would also have in addition to symmetries h_1 and h_2 — symmetries h_3 and h_4, for this 8×8 block must be necessarily present also symmetries h_5 and h_6 and each of its subblocks 4×4 and 2×2 must be positive for the presence of symmetries h_8 and h_9 too. With the appropriate combination of other symmetries within the individual subblocks 4×4 and 2×2, we could thus reach the limit values of the L characteristic.

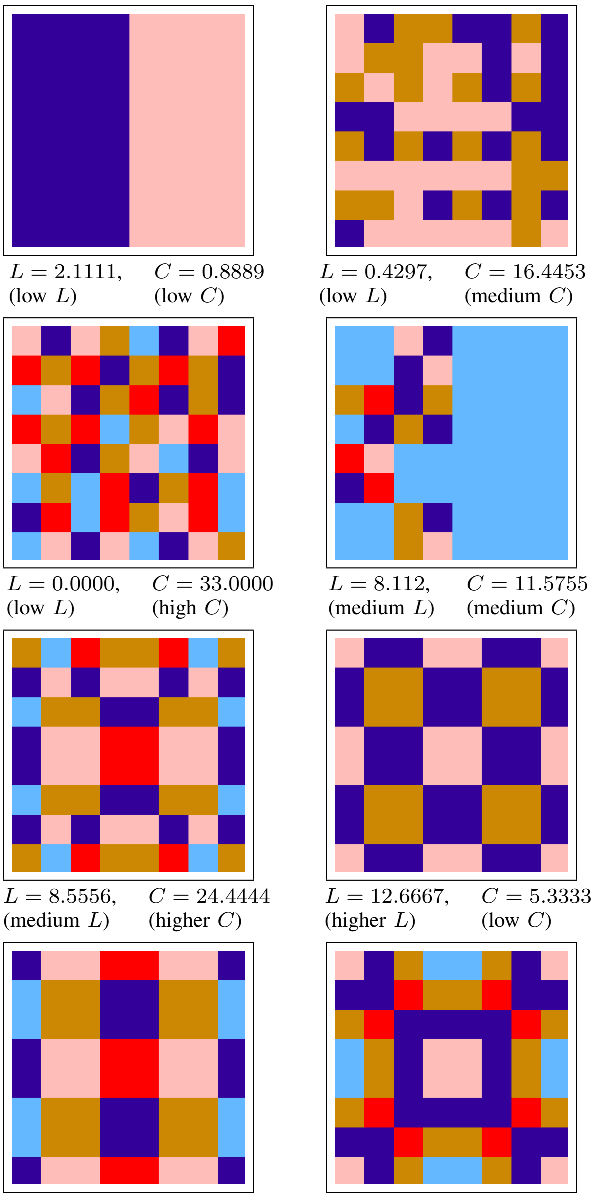
Taking into account equation (1) for calculating characteristics L, it is obvious that the aim is not to find an image with the maximum number of symmetries present. The value of the H characteristic, depending on the number of symmetries, will be maximal for such an image; but the T characteristic, that measures diversity elements, will be zero. This will also cause a zero value of the L characteristic of this image.

Because, the characteristic L is directly proportional to the product of values T and H, to find its maximum, we will need to find the values of characteristics T and H such that their product is the maximum!

To achieve symmetries h_1 and h_2 for block 8×8 in the image, it will be enough to generate only one quadrant of the image, which will then be duplicated, and according to the rules of axial symmetry, successively flipped over into the other quadrants. With this substantial simplification, we reduce the number of combinations, needed to generate up to four-color images, to 4^{16} (which is approximately 4 billion). Such a number of images can be generated and processed within a few tens of hours. For practical reasons, only images with unique values of L and C were saved when generating.

E. Summary and comparison of the results.

Now we will introduce several generated images (symmetric patterns are chosen in particular) to allow the reader to monitor dependency of aestetic characteristics and the patterns.



$L = 2.1111$, $C = 0.8889$
(low L) (low C)

$L = 0.4297$, $C = 16.4453$
(low L) (medium C)

$L = 0.0000$, $C = 33.0000$
(low L) (high C)

$L = 8.112$, $C = 11.5755$
(medium L) (medium C)

$L = 8.5556$, $C = 24.4444$
(medium L) (higher C)

$L = 12.6667$, $C = 5.3333$
(higher L) (low C)

$L = 12.2222$, 20.7778
(higher L) (higher C)

$L = 16.8889$, $C = 11.6111$
(high L) (medium C)

Fig. 11. Images with different values of L and C characteristics.

In Fig. 12 we will show the position of all the generated images. The L, C chart shows 300 million randomly generated images, 100 million images created by patterns combining, 13 200 images created by using a genetic algorithm, 780 images created manually, and images generated to achieve high symmetries (with h_1 and h_2 symmetry for block 8×8). A graph of absolute occurrence frequencies is displayed below.

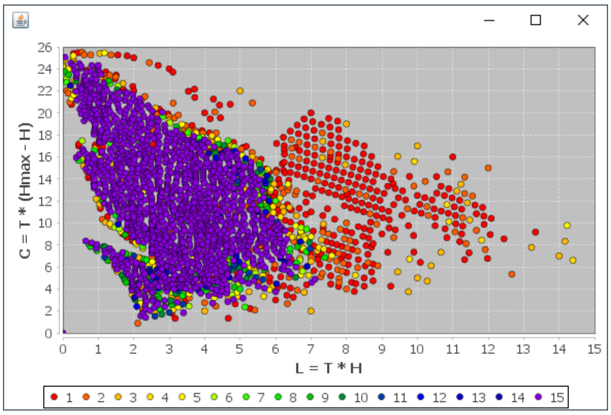


Fig. 12. Graph of L and C values of 300 million randomly generated images, 100 million images created by pattern combining, 13 200 images created by using genetic algorithm and 780 images created manually, and images generated to achieve high symmetries (containing h_1 and h_2 symmetry for block 8×8). A graph of absolute occurrence frequencies is used to display this.

Using these several methods, the L and C function value range was almost found for two, three, four, and five-color images.

Now we wrap the value ranges with convex envelopes (see Fig. 13) and calculate their individual centers of gravity (cf. Tab. I), which we then use for further calculations.

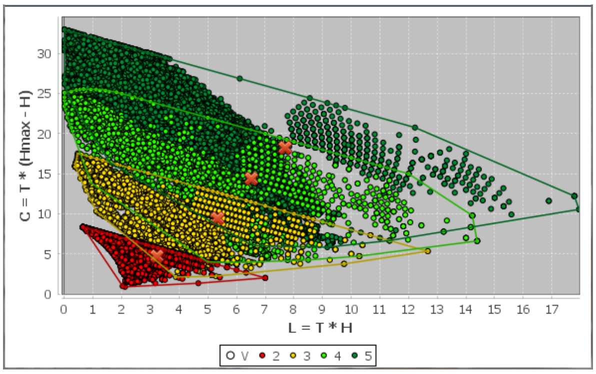


Fig. 13. Values of L and C characteristics for all acquired images.

In Tab. I, we can compare the focus of the range of image group values which dependend on the method used to generate them. This can also be compared using the values in Fig. 5, 7, 9, 12. From Tab. I it is obvious that the center of gravity of the RG method has the smallest L coordinates and the largest coordinates of C. Envelopes for image groups generated by the PC and GA methods have a centers of gravity with a larger L coordinate than the RG method, but a minor C coordinate. The image group envelope generated by the HS method has the center of gravity with the highest value L.

TABLE I
CENTERS OF GRAVITY OF THE L AND C RANGES.

method	**Colors**			
	2	3	4	5
random generation	[1.71,6.30]	[1.68,13.77]	[1.45,20.62]	[1.32,26.82]
pattern combining	[2.41,4.22]	[2.86,10.08]	[3.55,14.04]	[3.92,18.77]
genetic generation	[3.23,3.74]	[3.61,9.13]	[3.92,13.99]	[4.73,19.14]
high symmetries	[4.34,3.09]	[5.44,8.33]	[6.64,13.33]	[7.70,17.20]
all	[3.25,3.70]	[5.37,8.36]	[6.55,13.35]	[7.70,17.20]

III. Conclusion

In the first instance we tried to prepare an exactly described algorithm for calculation of the aesthetic characteristics of Life and Complexity. We complement the cited literature that did not fulfill this goal. The main aim of this paper was to identify the value range of L and C characteristics. For this purpose, we used several methods of generating images – generating by random algorithm, pattern combining method, generating images using genetic algorithm, and manual image creation with content of specific types of symmetry. We have managed to find a significant part of the value range for two to five color images. We explored the fact, that the value range of L and C characteristics depend on number of colors. Klinger and Salingaros have formulated a hypothesis that claims that with varying aesthetics characteristics, the audience's emotions also change. In order to be able to test this hypothesis on a random sample, we need to know, not only the field of aesthetic characteristics, but also their center of gravity. We found that aesthetic characteristics were conditional on the number of colors. This fact was not previously known in the literature. If the hypothesis of Klinger and Salingaros is accepted, then the emotional dependence on aesthetic characteristics will be described. Based on such knowledge, an image, that produces the greatest possible emotion (e.g. which viewer energises, excites, reassures, etc.) could be traced.

Acknowledgment

This research was supported by the Internal Grant Agency of University of Pardubice, the project SGS_2017_024.

References

[1] A. Klinger and N. A. Salingaros. "A Pattern Measure." *Environment and Planning B: Planning and Design*, vol. 27, no. 4, pp. 537-547, Aug. 2000. DOI: doi:10.1068/b2676. ISSN 0265-8135. Available: https://arxiv.org/html/1108.5508v1

[2] A. Garrido. "Symmetry and Assymetry Level Measures." *Symmetry*, vol. 2, pp. 702-721, Aug. 2010. DOI: doi:10.3390/sym2020707.

[3] Majid al-Rifaie, M. and A. Ursyn, R. Zimmer, M.A. Javaheri Javid, "On Symmetry, Aesthetics and Quantifying Symmetrical Complexity" *In J. Correia et al. (Eds.): EvoMUSART 2017, LNCS 10198*, pp. 1732, Apr. 2017. DOI: 10.1007/978-3-319-55750-2_2

[4] den Heijer, E. and A.E. Eiben. Investigating aesthetic measures for unsupervised evolutionary art. *Swarm and Evolutionary Computation*. vol. 16, pp. 52-68, Jun. 2014. DOI: 10.1016/j.swevo.2014.01.002. ISSN 22106502.

[5] Chen, Z. and M. Dehmer, F. Emmert-Streib, A. Mowshowitz, Y. Shi. "Toward Measuring Network Aesthetics Based on Symmetry." *Axioms*. vol. 6, no. 2, p. 12, May 2017. DOI: doi:10.3390/axioms6020012.

[6] Russell, S. and P. Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009. ISBN 978-0136042594.

Substitution Steganography with Security Improved by Chaotic Image Encryption

Jakub Oravec and Ján Turán
Dept. of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Park Komenského 13, 040 01 Košice, Slovakia
jakub.oravec@tuke.sk

***Abstract*—This article describes an information hiding scheme, which combines advantages of steganography and cryptography. After brief review of currently used techniques and their properties, a chaotic image encryption method based on quadratic map is proposed. This algorithm is then utilized as a tool for encryption of image representing secret data, which is afterwards embedded into cover image by means of LSB matching. Paper also provides verification of results achieved by both parts of presented approach and discusses its advantages and drawbacks.**

***Keywords*—chaotic map, image encryption, Least Significant Bit matching, Quadratic map, steganography**

I. Introduction

Techniques of information hiding have been evolving for a long time. *Steganography* could be considered as one of their earliest examples [1]. Nowadays, steganography is utilized for exchange of secret data hidden in various data files. These files called covers are modified in a way that should not be easily recognized, but informed users should be able to extract hidden data. Therefore, security of this solution is based mainly on fact that other users do not suspect usage of steganography.

On the other hand, *cryptography* is not used for hiding the presence of secret data. Instead, it tries to modify meaning of the data by performing steps of encryption algorithm. In most cases, these steps depend also on set of parameters called key.

Development of new steganographic algorithms created also new ways of steganalysis. One of popular steganalysis tools is based on comparing values of features calculated for cover data with embedded secret data (known as stego data) and the original cover data. An example set of these features can be found in [2] where 274 features were used. As amount of these features expanded over thousands [3], [4], some authors decided to provide other solution than improvement of steganographic algorithms properties [5]–[7].

The other solution is based on combination of steganography and cryptography. While steganography is used for embedding of secret data, cryptography hides its meaning prior to embedding. Therefore, if usage of steganography is revealed, it is not possible to get the original secret data until key is obtained. This approach enables usage of simple steganographic techniques operating with least significant bits (LSBs) of data, such as LSB replacement, or LSB matching.

However, these techniques are usually combined with conventional encryption algorithms, such as Advanced Encryption Standard (AES) [7]–[9]. In most cases, these algorithms were developed for encrypting text strings rather than images [10], which are usually used for steganography. Because images have different properties, e.g. many pixels with largely correlated values [11], the encryption results are not sufficient in all cases. Also amount of their operations can be lowered without causing big differences in properties of encrypted images [12].

A possible solution of this problem is usage of chaotic ciphers which were introduced by Matthews [13]. A two stage architecture of chaotic image encryption was proposed by Fridrich [14]. First step, called *confusion* shuffles image pixels and thus it breaks correlation among adjacent pixels. Second step, *diffusion* is used for modification of amplitudes of image pixels. Usage of diffusion also prevents some of known attacks.

The rest of paper is organized as follows: second chapter briefly describes principle of LSB matching steganography and properties of quadratic map. The algorithms for encryption and steganography are explained in third chapter. Experimental results and discussion are present in fourth chapter. Paper concludes in fifth chapter with topics for possible future work.

II. Used Techniques

A. LSB matching

Steganographic technique of LSB matching was developed as an improvement of LSB replacement [15]–[17]. Both approaches exploit redundancy in grayscale cover images. The difference between them is in a way of secret data embedding. While LSB replacement directly replaces LSBs of cover pixel amplitudes with bits of secret data, LSB matching can use two different amplitudes for each secret data bit.

The amplitude of stego image pixel in case of LSB matching is changed only when its LSB is not equal to bit of secret data. In this case, it is either incremented or decremented, the decision usually depends on values of generated pseudorandom sequence (PRS).

Exception is present for pixels with saturated amplitudes (0 or 255). These values could produce amplitudes which exceed range of grayscale values. Usually, LSB replacement is used for embedding of secret data into these pixels [16], [17].

B. Quadratic map

Functions, which were later modified to create quadratic map (QM) were studied for first time during early 1970s [18]. QM can be described as one dimensional chaotic map which maps real numbers from range $\langle -2; 2\rangle$ to the same range [19].

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

It uses initial value x_n and one parameter $a \in \langle 0; 2\rangle$ for calculations of successive iterates x_{n+1}. The equation for QM is given as (1):

$$x_{n+1} = a - x_n^2, \tag{1}$$

where $n = 1, 2, \dots$ denotes iteration number.

Chaotic properties of QM can be illustrated by bifurcation diagram, shown on Figure 1. The behavior of QM is predictable when a is less than $\sim$ 0.74249. At this point, first bifurcation takes place. Values of x_n oscillate between two sequences until next bifurcation ($a \sim$ 1.24749) is reached. After this point, newly computed x_n achieve values from four sequences. As the number of these possible sequences reaches infinity, the so-called *chaos* is introduced [20]. However, certain values of a, e.g. around 1.75 still produce islands of stability, where the behavior of map is predictable [21].

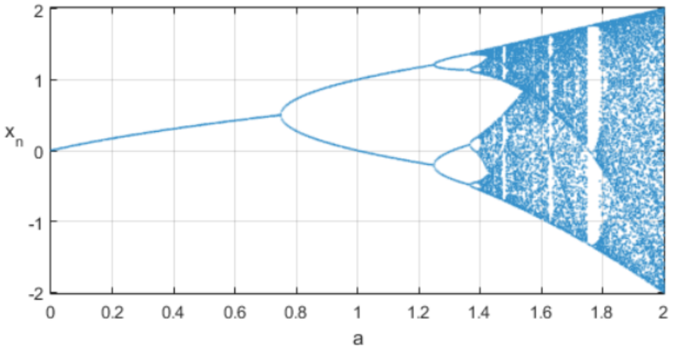


Figure 1. Bifurcation diagram of quadratic map.

III. Proposed Solution

In our solution, we tried to combine advantages of steganography and cryptography for providing certain level of security for an information hiding technique. This feature can be achieved by encrypting secret data prior to its embedding by means of steganography. At the receiving side, extracted secret data is decrypted and the original meaning of secret data is available to user. A block scheme of these operations is illustrated on Figure 2.

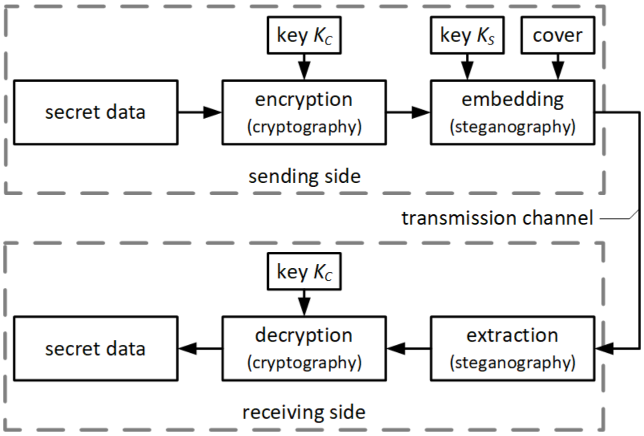

key K_C
key K_S
cover
secret data
encryption (cryptography)
embedding (steganography)
sending side
transmission channel
key K_C
secret data
decryption (cryptography)
extraction (steganography)
receiving side

Figure 2. A block scheme of operations done in proposed solution.

Please note that some steganographic algorithms do not use key for extraction which is also the case of proposed solution.

A. Image Encryption by Quadratic map

Encryption and decryption of secret data in form of grayscale images is done by algorithms based on QM. Their architecture is similar to those proposed by Fridrich [14]. The algorithms are used for four times with different keys in order to reach keyspace comparable with 128-bit AES.

Firstly, image containing secret data is padded until amount of its bits reaches the capacity provided by cover image. This step ensures that padding pixels will be encrypted. Hence, the encrypted image would have uniform appearance which should be similar to LSB bitplane of original cover image.

Then *confusion* shuffles image pixels in two steps, first one changes position of pixels in rows of image and the second step moves pixels in image columns. Each iteration of confusion takes two eight bit words as parameters. These words are converted to decimal numbers, divided by 100 and then added to value of 1.99. This approach generates values ranging from 1.99 to 2 which are afterwards used as parameters for QM.

The values of shifts in image rows sh_r and image columns sh_c are computed by iterating QM. First 100 iterates are thrown away for reaching sufficient chaotic behavior of generated sequences. Then following h or w iterates are passed to sh_r and sh_c respectively, where h denotes height of image and w is its width.

At this point, values of shifts sh_r and sh_c are from range $\langle -2; 2\rangle$. They are quantized by using (2):

$$val_q = round((dim - 1) \cdot \frac{val + 2}{4}), \tag{2}$$

where val_q denotes quantized value, dim is maximal possible value of shift (h for values from sh_c and w for values from sh_r) and val is actual value from sh_c or sh_r.

The effects of one iteration of confusion are displayed on Figure 3. Keys used in this case were 10010100 for shifting done in image rows and 11001110 for image columns shift.

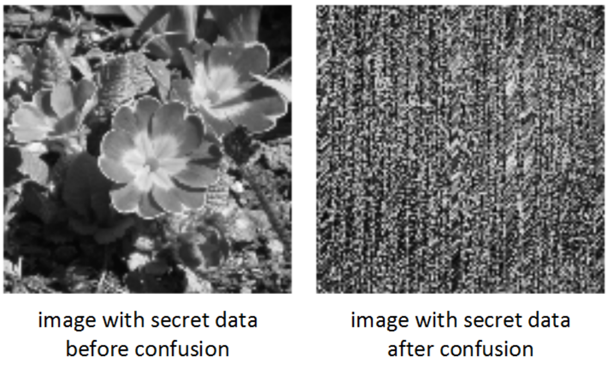


image with secret data before confusion
image with secret data after confusion

Figure 3. An example of image before and after confusion stage.

Second step of encryption algorithm is *diffusion*. Here, the amplitudes of shuffled image pixels are modified. The operations which are done provide certain level of robustness against statistical and differential attacks.

Diffusion stage uses QM to produce a sequence of numbers in similar fashion like the confusion step. However, there are several differences. The key length is in this case 16 bits and computed elements are quantized to a set $\{0, 1, \ldots, 255\}$.

At first amplitudes of shuffled pixels are reshaped into a vector. Then this vector is processed by two iterations of (3):

$$enc_n = (pl_n + enc_{n-1} \ (mod \ 256)) \oplus sh_{a,n}, \qquad (3)$$

where enc_n denotes nth element of vector with encrypted data, $n = 1, 2, \ldots, h \cdot w$, h and w are image height and width, pl is vector with plaintext, $\oplus$ represents operation of bitwise exclusive or (XOR) and sh_a denotes generated PRS. Plaintext is represented by shuffled image for first iteration of diffusion and by vector calculated after first iteration in second iteration. First element done in first iteration of diffusion (enc_1) does not use previous element (enc_0) because it does not exist.

Above mentioned equation can be broken down to two operations. First one – addition of previous element, known as *ciphertext chaining* helps to spread differences between various images to all pixels of resulting encrypted images. Thanks to two iterations of diffusion, even difference in one pixel produce different encrypted images. Second operation, bitwise XORing with PRS generated by chaotic map causes properties, which are desired in order to suppress chance of successful attacks.

The effects of one iteration of diffusion are illustrated on Figure 4. Used key had value of 1110010001011100.

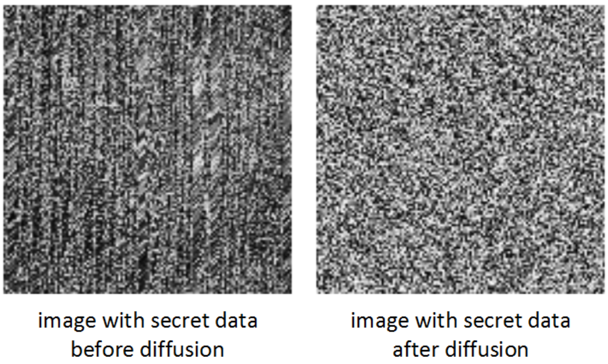


Figure 4. Effects of diffusion stage.

Finally, bytes of encrypted image are decomposed to corresponding bits as embedding by LSB matching needs secret data in form of binary vector.

Decryption is analogous to encryption, however the inverse diffusion is done prior to inverse version of confusion. As decryption algorithm uses the same procedure for calculation of pixel and amplitude shifts, it requires the same set of keys.

In the case the image with secret data was padded before encryption, decryption produces image with pixels added by padding. Algorithms, which could suppress this drawback are good topic for our future work.

B. Secret Data Embedding by LSB matching

Encrypted images containing secret data are embedded into grayscale cover images by means of LSB matching. If cover bytes are not saturated, they are modified according to PRS generated by QM.

This PRS is generated in four steps. Each step utilizes part of key with length of 16 bits. First three steps are used for generating initial value for following step. Desired chaotic behavior is achieved by throwing away first 100 iterates. Then, next iterate is used by following step. Finally, fourth step uses initial value generated by third step, also throws away first 100 iterates and then computes another num_S iterates, where num_S represents number of secret data bits.

The bits of secret data are then embedded by Algorithm 1:

Algorithm 1: LSB matching – embedding

```
Input  : bit of secret data $m$, byte of cover data $C$ and
         element of pseudo-random sequence $e$
Output: byte of stego data $S$
if $C \ (mod \ 2) \ != m$ then
    case $C == 0$ && $m == 1$
        $S = C + 1;$
    end
    case $C == 255$ && $m == 0$
        $S = C - 1;$
    end
    otherwise do
        if $e > 0$ then
            $S = C + 1;$
        else
            $S = C - 1;$
        end
    end
else
    $S = C;$
end
```

Operation inverse to embedding, extraction is easier as both saturated and not saturated pixels have LSBs that are equal to bits of embedded secret data. As extraction algorithm does not have to choose the way of pixel modification, it does not need the key used for generating PRS. The steps of extraction are described in Algorithm 2:

Algorithm 2: LSB matching – extraction

```
Input  : byte of stego data $S$
Output: bit of secret data $m'$
if $S \ (mod \ 2) == 0$ then
    $m' = 0;$
else
    $m' = 1;$
end
```

IV. Experimental Results

The properties of our solution were verified by series of experiments conducted in software pack Matlab R2015a. Used PC was running Windows 10, it had 2.5 GHz CPU and 12 GBs of RAM. Cover images had resolution of 512x512 pixels and color depth of 8 bits. The images with secret data had resolution of 128x128 pixels with the same color depth.

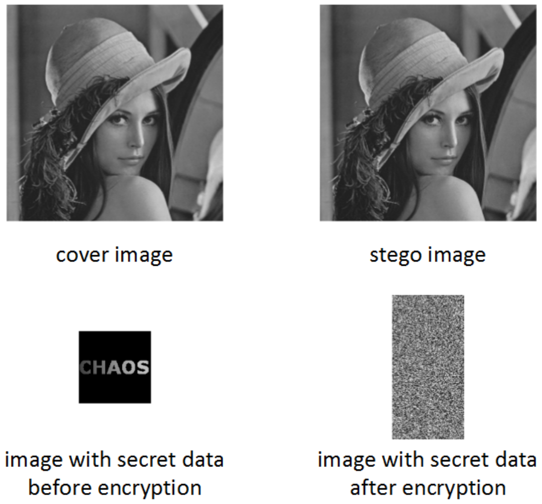
A. Encryption Key Sensitivity

Experiments used keys in form of binary vectors. These keys are listed in Table I in hexadecimal notation for sake of shortness. Conversion was done by big-endian ordering scheme. Bold font indicates differences between the keys.

Table I. KEYS USED DURING EXPERIMENTS

key	keys used during shifts	key used for generating PRS
K_1	0x94959698, 0xA208D677 0x1523B9ED0FAA3082	0xF3285ACEF143605B
K_2	0x949596**97**, 0xA208D677 0x1523B9ED0FAA3082	
K_3	0x94959698, 0xA208D677 0x1523B9ED0F**AB**3082	

Effects of proposed algorithms are displayed on Figure 5. This example uses key K_1. Please note that the secret data had to be padded to reach the capacity of cover before encryption.


cover image

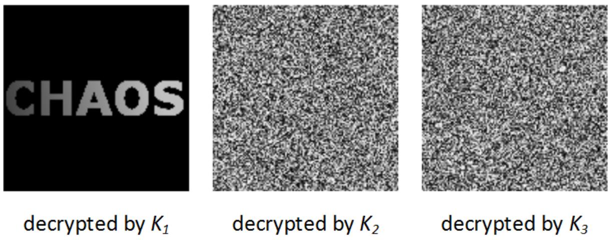
stego image

image with secret data before encryption

image with secret data after encryption

Figure 5. Comparison of cover and stego images.

Results of decryption with incorrect keys are shown on Figure 6. Image with secret data was encrypted by K_1. Keys K_2 and K_3 produce different parameters for QM, so shifts applied during confusion or diffusion are also other. Decrypted images are magnified in order to display more details.


decrypted by K_1

decrypted by K_2

decrypted by K_3

Figure 6. Images decrypted by various keys.

As it can be seen, the last two images do not resemble correctly decrypted image. This is caused by four iterations of decryption, where even slightly shifted version of desired image produces different image in next decryption iteration.

B. Similarity of Cover and Stego Images

Differences between cover and stego images could be examined by several measures [22]. In this paper we use values of Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM). First mentioned measure is computed by (4) and (5):

$$PSNR = 10 \cdot log_{10} \frac{2^L - 1}{MSE} \; [dB], \tag{4}$$

$$MSE = \frac{1}{h \cdot w} \sum_{l=1}^{h} \sum_{k=1}^{w} (S_{l,k} - C_{l,k})^2, \tag{5}$$

where L is color depth of images (8 bits for grayscale images), MSE abbreviates Mean Squared Error, h and w are height and width of images, S is stego image, C is original cover image, l and k are indices of image rows and columns.

Second measure, SSIM is calculated by using (6)–(10):

$$SSIM = \frac{(2 \cdot \mu_S \mu_C + c_1) \cdot (2 \cdot cov_{SC} + c_2)}{(\mu_S^2 + \mu_C^2 + c_1) \cdot (\sigma_S^2 + \sigma_C^2 + c_2)}, \tag{6}$$

$$\mu_X = \frac{1}{h \cdot w} \sum_{l=1}^{h} \sum_{k=1}^{w} X_{l,k}, \tag{7}$$

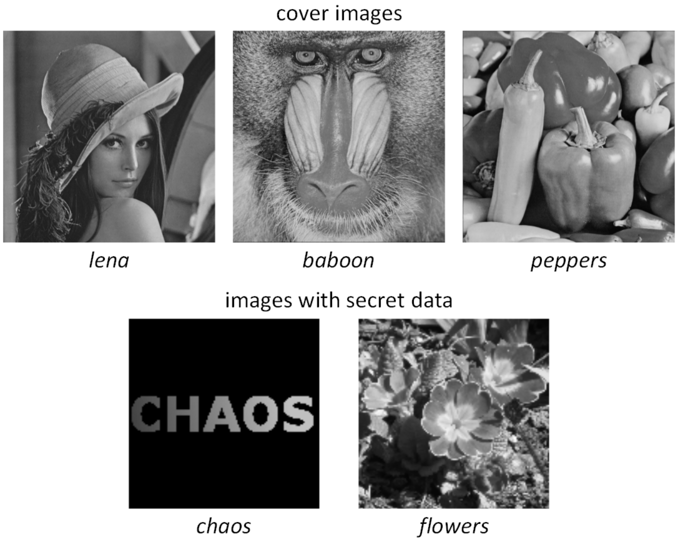
$$cov_{XY} = \sum_{l=1}^{h} \sum_{k=1}^{w} (X_{l,k} - \mu_X) \cdot (Y_{l,k} - \mu_Y), \tag{8}$$

$$\sigma_X^2 = \sum_{l=1}^{h} \sum_{k=1}^{w} (X_{l,k} - \mu_X)^2, \tag{9}$$

$$c_1 = 0.01 \cdot (2^L - 1), \; c_2 = 0.03 \cdot (2^L - 1), \tag{10}$$

where μ_X is average of image X, σ_X^2 is its variance and cov_{XY} is covariance of images X and Y.

Images used in comparison are displayed on Figure 7. The images containing secret data are magnified.


cover images

lena

baboon

peppers

images with secret data

chaos

flowers

Figure 7. Reference images used in computation of PSNR and SSIM.

Following PSNR and SSIM calculations used key K_1. The resulting values are shown in Table II.

Table II. COMPUTED VALUES OF PSNR AND SSIM

cover/ secret data	*lena*		*baboon*		*peppers*	
measure	PSNR [dB]	SSIM	PSNR [dB]	SSIM	PSNR [dB]	SSIM
chaos	51.1743	0.9962	51.1726	0.9987	51.1912	0.9963
flowers	51.1768	0.9962	51.1712	0.9987	51.1682	0.9963

The values of PSNR and SSIM calculated for two images with secret data show that encryption produces similar encrypted images for various input images. PSNR values are higher when encrypted image is more identical to LSB bitplane of original cover image.

V. Conclusion

This paper described a combination of chaotic image encryption algorithm and steganographic technique known as LSB matching. While encryption is used for preprocessing of secret data in form of grayscale image, steganography is utilized for embedding of encrypted secret data into a cover image. The security of this solution thus depends on detection of steganography usage and set of keys utilized during the encryption.

Mentioned approach has two advantages over steganographic systems which do not use encryption. Firstly, if secret data is acquired via a successful attack, it is still encrypted, so its meaning is not clear without decryption. Secondly, encrypted secret data should resemble LSB bit plane of original cover image, because both these images should look like noise. This fact is also confirmed by achieved values of SSIM.

The encryption algorithm is based on two operations which exploit properties of chaotic map. As this algorithm is proposed specifically for this particular application, it reaches desired effects with minimal amount of computations.

One visible drawback is caused by unknown size of secret data during extracting step. Therefore, secret data had to be padded before they were encrypted and embedded into cover data. This method leads to extraction of secret data in form of image with visible padding, if size of secret data is not equal to capacity provided by cover data. Possible solution of this problem is a good topic for our future research. One approach is usage of header which would contain information about the size of secret data. However, the embedding of this header into cover data would decrease capacity for secret data itself.

Acknowledgment

This work was supported by following research grants: KEGA 023TUKE-4/2017 (New Trends in the Optical Data Transmission), VEGA 1/0772/17 (Multiple Person Localization Based on Detection of Their Vital Signs Using Short-Range UWB Sensors) and ITMS 26220120055.

References

[1] S. Singh, *The Code Book*, London: HarperCollins Publishers, 2000. 416 p. ISBN: 978-18-5702-889-8.

[2] T. Pevný, J. Fridrich, "Merging Markov and DCT Features for Multi-Class JPEG Steganalysis," *Proc. of SPIE*, vol. 6505, San Jose (USA), 2007, p. 1–13. ISBN: 978-08-1946-618-1. DOI: 10.1117/12.696774.

[3] J. Kodovský, J. Fridrich, "Steganalysis in high dimensions: fusing classifiers built on random subspaces," *Proc. of SPIE*, vol. 7880, San Jose (USA), 2011, p. 1–13. ISBN: 978-08-1948-417-8. DOI: 10.1117/12.872279.

[4] J. Fridrich, J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, 2012, vol. 7, no. 3, p. 868–882. ISSN: 1556-6013. DOI: 10.1109/TIFS.2012.2190402.

[5] V. Bánoci, G. Bugár, D. Levický, Z. Klenovičová, "A Novel JPEG Steganography Method Based on Modulus Function with Histogram Analysis," *Radioengineering*, 2012, vol. 21, no. 2, p. 758–763. ISSN: 1805-9600.

[6] J. K. Saini, H. K. Verma, "A Hybrid Approach for Image Security by Combining Encryption and Steganography", *Proc. of 2nd Intl. Conf. ICIIP 2013*, Waknaghat (India), 2013, p. 607–611. ISBN: 978-14-6736-101-9. DOI: 10.1109/ICIIP.2013.6707665.

[7] V. Hajduk, M. Broda, O. Kováč, D. Levický, "Image Steganography with Using QR Code and Cryptography," *Proc. of 26th Intl. Conf. Radioelektronika 2016*, Košice (Slovakia), 2016, p. 350–353. ISBN: 978-15-0901-673-0. DOI: 10.1109/RADIOELEK.2016.7477370.

[8] S. F. Mare, M. Vladutiu, L. Prodan, "Secret Data Communication System Using Steganography, AES and RSA," *Proc. of 17th Intl. Symposium SIITME 2011*, Timisoara (Romania), 2011, p. 339–344. ISBN: 978-14-5771-275-3. DOI: 10.1109/SIITME.2011.6102748.

[9] A. Conci, A. L. Brazil, S. B. L. Ferreira, T. MacHenry, "AES Cryptography in Color Image Steganography by Genetic Algorithms," *Proc. of 12th Intl. Conf. AICCSA 2015*, Marrakech (Morocco), 2015, p. 1–8. ISBN: 978-15-0900-478-2. DOI: 10.1109/AICCSA.2015.7507100.

[10] W. Steingartner, D. Radaković, F. Valkošák, P. Macko, "Some properties of coalgebras and their rôle in computer science," *J. of Applied Mathematics and Computational Mechanic*, 2016, vol. 15, no. 4, p. 145–156. ISSN: 2353-0588. DOI: 10.17512/jamcm.2016.4.16.

[11] O. Kováč, T. Girašek, A. Pietriková, "Image Processing of Die Attach's X-Ray Images for Automatic Voids Detection and Evaluation," *Proc. of 39th Intl. Spring Seminar on Electronics Technology ISSE 2016*, Plzeň (Czech Republic), 2016, p. 199–203. ISBN: 978-15-0901-389-0. DOI: 10.1109/ISSE.2016.7563188.

[12] W. Steingartner, V. Novitzká, "Categorical model of structural operational semantics for imperative language," *J. of Information and Organizational Sciences*, 2016, vol. 40, no. 2, p. 203–219. ISSN: 1846-9418.

[13] R. Matthews, "On the Derivation of a 'Chaotic' Encryption Algorithm," *Cryptologia*, 1989, vol. 8, no. 6 p. 29–41. ISSN: 0161–1194. DOI: 10.1080/0161-118991863745.

[14] J. Fridrich, "Symmetric Ciphers Based on Two-dimensional Chaotic Maps," *Intl. J. of Bifurcation and Chaos*, 1998, vol. 8, no. 6, p. 1259–1284. ISSN: 0218–1274. DOI: 10.1142/S021812749800098X.

[15] A. Ker, "Improved detection of LSB steganography in grayscale images," *Proc. of 6th Information Hiding Workshop*, vol. 3200, Toronto (Canada), 2004, p. 97–115. ISBN: 978-35-4030-114-1.

[16] J. Mielikainen, "LSB matching revisited," *IEEE Signal Processing Letters*, 2006, vol. 13, no. 5, p. 285–287. ISSN: 1070-9908. DOI: 10.1109/LSP.2006.870357.

[17] J. Fridrich, *Steganography in Digital Media: Principles, Algorithms and Applications*, Cambridge: Cambridge University Press, 2009. 466 p. ISBN: 978-05-2119-019-0.

[18] A. V. Aho, N. J. A. Sloane, "Some Doubly Exponential Sequences," *Fibonacci Quarterly*, 1973, vol. 11, no. 4, p. 429–437. ISSN: 0015-0517.

[19] F. J. S. Moreira, "Chaotic dynamics of quadratic maps," Master's thesis, University of Porto (Portugal), 1992, 50 p.

[20] J. Gleick, *Chaos*, London: Vintage Books, 1998. 352 p. ISBN: 978-07-4938-606-1.

[21] R. May, "Simple mathematical models with very complicated dynamics," *Nature*, 1976, vol. 261, no. 5560, p. 459–467. ISSN: 0028-0836. DOI: 10.1038/261459a0.

[22] A. Samcović, J. Turán, "Attacks on Digital Wavelet Image Watermarks," *J. of Electrical Engineering*, 2008, vol. 59, no. 3, p. 131–138. ISSN: 1335-3632.

Deployment of the PON with an Optical Fibre G-652.B

Ľuboš Ovseník, Ján Turán, Tomáš Ivaniga
Department of Electronics and Multimedia Communications
University of Technology Košice
Košice, Slovakia
lubos.ovsenik@tuke.sk, jan.turan@tuke.sk,
tomas.ivaniga@tuke.sk

Petr Ivaniga
Department of Information Networks
Faculty of Management Science and Informatics
University of Žilina
Žilina, Slovakia
petr.ivaniga@fri.uniza.sk

***Abstract*— The aim of this article was to create two passive optical networks PON (Passive Optical Network). In the LOS (Laboratory of Optoelectronic Systems) at KEMT (Department of Electronics and Multimedia Communications), FEI (Faculty of Electrical Engineering and Informatics), TUKE (Technical University of Košice) were created the optical networks of different lengths. For implementation was used the optical fibre G-652.B with a length of 706.4m, PLC (Planar Lightwave Circuit) splitter with a splitting ratio of 1:4, connectors SC/PC, and RISER indoor switchgear. The optical networks were first designed, and afterwards simulated and implemented. Design and simulation were created in OptSim software from Rsoft company. The main contribution of this paper was comparison of the difference between simulated and measured parameters of optical networks and demonstration when it is better to use an optical splitter, and when it is better to connect the fibre directly. The distances were set by the buildings based at FEI which were UVT (Institute of Computer Science) and the LOS laboratory at V4 (Vysokoskolska 4).**

Keywords*— *FTTH, OptSim, PLC splitter, PON

I. Introduction

In the last decade the optical communications systems became well discussed topic among domestic and foreign authors. The components of optical systems were theoretically analyzed, and in practice effectively used particularly in the sphere of telecommunications. The use of optical fibre and sources of coherent light were a great motivation for many scientists to develop optical communication systems [1], [3].

The optical fibres are nowadays considered as an efficient transmission medium designed to transmit a large amount of data over a long distance. Compared to other transmission media (e.g. a free space and copper cables) they have exceptional qualities. The optical networks are divided by way of sharing used fibres and network termination units to point-to-point (P2P) and point-to-multipoint (P2MP) networks.

For the P2P optical networks is transmission path and each communication unit at the end side intended for one end user. For P2MP optical networks is part of the optical infrastructure including central communication units shared by a greater number of end users.

II. Description of PON, FTTx and PLC splitter

This chapter describes PON, FTTx with a focus on FTTH and PLC splitter used in the implementation.

A. Structure of PON

The first steps in PON technology led to the idea of transmitting data signals through a single optical fibre to multiple end users. Passive optical access networks belong to a group of multipoint access networks with a passive ODN (Optical Distribution Network) and reduce the amount of necessary optical fibre and equipment compared to the point-to-point network. The main task is to connect a large number of end users in a given area. Current PON networks use time multiplex – TDM for optical fibre sharing, and the transfer speed is shared among all active end users.

The designations ONU (Optical Network Unit) and ONT (Optical Network Termination) usually coincide in practice, and many providers and manufacturers of optical components do not distinguish these two different concepts [1], [2] and [4]. The control of end units ONU and ONT runs remotely from the optical central unit OLT (Optical Line Termination), which is in charge of the whole PON. In Fig.1 are schematically shown the basic elements of a passive network and the difference between the ONU and ONT. Network equipment NT (Network Termination) is the end point-binding of xDSL (x Digital Subscriber Line), Ethernet etc.

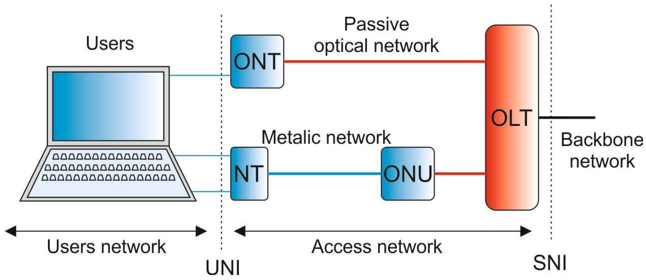


Fig. 1. Structure of PON with ONT and ONU.

B. FTTx and FTTH

1) FTTx

The construction of new access networks began to develop the idea of bringing optical fibre to the end user as close as possible. The structure of the optical access network is not in any case purely optical. In order to achieve lower costs in building optical infrastructure we can partially use metallic lines and networks [4], [5]. The concept of hybrid optical-copper connections, which are a combination of optical network and downstream copper infrastructure, is called FTTx (Fibre To The x).

2) FTTH

FTTH is a pure optical network in which the optical fibre is brought to the end user without external distributors, coils and metallic lines. Great advantage of this alternative, same as in methods FTTO (Fibre To The Office) and FTTD (Fibre To The Desk), is a high bit rate (from 50 to 100Mbit.s^{-1}) [3], [6]. This method of optical connection is financially the most challenging, taking into account the requirements for the construction of optical networks and bringing fibre to the end user. FTTH architecture finds its use especially in demanding multimedia services, and the distribution of television signals and HD videos.

C. PLC splitter

PLC splitters are made by planar technology. The first production step is to cut special plates from flint glass and subsequently create masks with a desired structure. The technological process creates the core of a waveguide, the coat of glass plate is the material itself. The input and output fibre arrays are produced afterwards, they are glass plates made of the same material. Into the plates are then cut so-called V-grooves, in which the individual fibre cores are inserted precisely. In the last step, the input and output optical fibres are connected. In terms of ensuring the adequate protection, the PLC splitter is sealed into a hard-protective case. PLC splitters with larger number of output ports (64, 128) are more difficult in accuracy, however the production technique is nowadays well controlled. The internal structure of the PLC splitter is shown in Fig.2 [6], [8]. Optical splitters are connected into ODN infrastructure via connectors, mechanical joints, or splices of optical fibres.

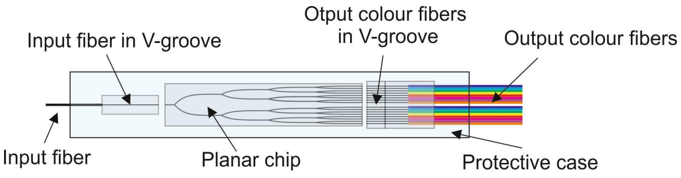


Fig. 2. PLC splitter.

III. OTDR

In terms of ensuring the reliability and maintenance of the transmission routes and optical fibres, it is very important to diagnose the optical fibres during their production and assembly process [7]. OTDR is also known as the backscatter method. The principle consists of broadcasting periodic-short optical pulses into analyzed optical fibres. Part of the optical signal is reflected back to the beginning of the fibre due to micro-inhomogeneities in the fibre core which is a result of linear Rayleigh scattering.

Fresnel reflection enables to locate inhomogeneities caused by e.g. incorrect connector connections or disconnected optical fibres. The result of the analysis is the backscatter curve on a logarithmic scale, which gives us information not only on the overall quality of the optical fibre depending on its length, but also information about each section of route [7]. In that case the fibre is homogeneous, the intensity of the reflected light decreases exponentially with a time due to attenuation in the fibre (Fig.3).

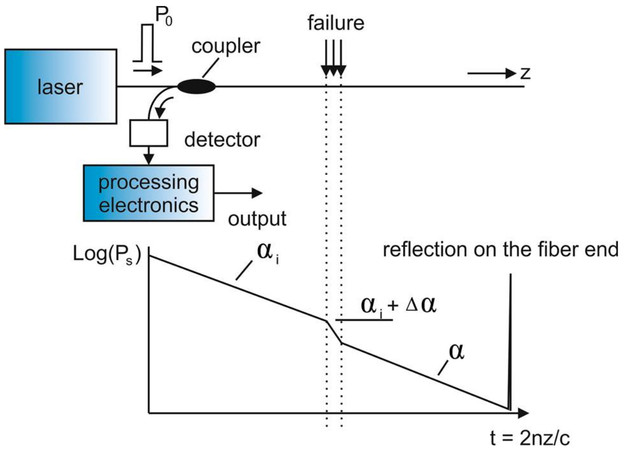


Fig. 3. The principle of OTDR measurement.

It is also possible to detect the insertion loss of connectors, splices, the total length of fibre faults and elements of the optical network [1], [3] and [8]. OTDR method has extensive use in all areas of optical communications. Measurement reveals an overall quality of the optical network [9]. The advantage is that the connection of optical reflectivity and measured network gives a sufficient access to only one end of the optical fibre. Maximum power of input pulse τ with a duration P_0, reflected power detected with a time delay t from sending the input pulse can be expressed as

$$P_s(t) = (1-\kappa)\kappa P_0 Dr(z) exp\left[-\int_0^z 2\alpha_i(z)dz\right], \qquad (1)$$

where $z=ct/2n$ is the position of the original pulse at the time of detected reflected signal $P_s(t)$, $\alpha_i(z)$ is the attenuation coefficient in Np (1 Np=8,68 dB), κ is the disengagement ratio of the input power, $r(z)$ is the coefficient of effective backscatter per length unit taking into account Rayleigh reflective coefficient and a cross-section of the fibre, $D=(c\tau/n)$ is the length of the light pulse in the fibre at any point in time, c is the speed of light and n is group index of the core optical fibre.

Assuming that the attenuations of the input pulse and the reflected light are equal, the slope of the logarithm of the detected signal is equal to a loss coefficient, which is expressed as

$$\frac{\partial(lnP_s)}{\partial z} = -2\alpha_i(z). \qquad (2)$$

In Fig.3 are areas with a high loss α_i shown by higher slope on the route OTDR. OTDR space resolution is defined by the smallest distance between two scatterings observed and determined by the width of the input pulse

$$\Delta z_{min} = \frac{c\tau}{2n} \qquad (3)$$

The pulse width of 10ns equals to OTDR resolution of about 1m. In general, the pulses transmitted by OTDR have low power and obtained results should be averaged to achieve a good signal to noise ratio. For measuring optical networks with a single mode fibre are nowadays used standard wavelengths of 1310nm and 1550nm [7], [12]. At a wavelength of 1310nm are significantly registered poor quality welds of optical fibres and incorrect connector connections. The wavelength of 1550nm is used for better diagnosis and localization of bends in order to identify potential faults in the optical path. Measuring parameters of optical fibres and optical routes by OTDR method is an important tool for ensuring the smooth operation of transmission networks.

IV. Implementation of Optical Networks

Prior to the implementation of the optical networks, it is necessary to simulate all phenomena that might affect the process [10], [11] and [13]. In LOS we have available software from RSoft Company which is used for simulations of real optical networks. PON networks are designed by this software. The whole implementation process of optical networks takes place in the following steps: optical networks design, simulation, execution of physical workplace, testing by using OTDR and final evaluation of the results.

A. *Design, implementation and simulation of optical networks*

The optical networks are designed between the buildings at TUKE. Individual buildings UVT and LOS are by GPS 1453m apart. Due to the floors in each building is the distance of 300m added to the optical fibre. Connectors SC/PC are used for the optical fibre termination in the RISER indoor switchgear and the attenuation of 0.16dB is in the simulation. Described implementation uses optical fibre G-652.B with a length of 706.4m and the attenuation of 0.4dB.km^{-1}. The optical splitter used in PON, has split ratio of 1:4 and the attenuation of 7.4dB. Individual optical welds (40mm) have the attenuation of 0.2dB. The series fibre is a part of the optical network during measurements, hence we count with the fibre in simulations. The optical fibre G-652.D has the attenuation of 0.33dB.km^{-1}. The topology in Fig.4 represents 2 passive optical networks. For simulations are used maximum attenuations of each component. The drawback of OptSim is the lack of optical network measurement by OTDR. The final spectrum at the output ranged from 193.362THz to 193.466THz. The attenuation of PON has a value of about 3dB and for usage PLC is the value about 10dB.

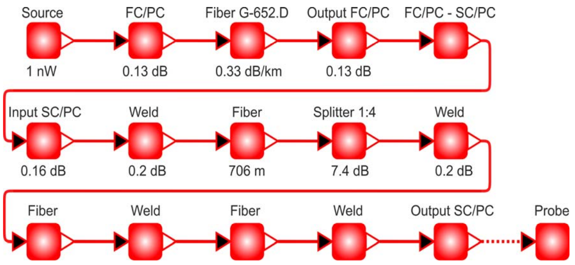


Fig. 4. Topology of optical networks by OptSim software.

B. *Control of final parameters by OTDR*

For the measuring of quality of optical networks the OTDR method is used for measuring optical fibres. In Fig.5 is measured the optical path PON and the final values are in TABLE I. The measurements are processed at a wavelength of 1310nm and 1550nm. The pulse width is 100ns and the measurement on a single line is averaged to 60s. Adjusted length measurement is 5000m. The optical fibre is at a length of 506m and connectors FC/PC are used to eliminate the dead zone.

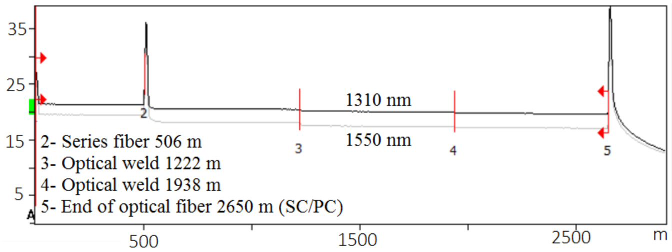


Fig. 5. The backscatter curve OTDR for PON at 1310nm and 1550nm.

TABLE I. Values for PON at 1310nm and 1550nm

No	Loc. (m)	Event Type	Loss (dB)	Refl. (dB)	Att. (dB/km)	Cum. (dB)
		1310nm				
1	0.0000	Launch Level	---	-48		0.000
		Sec. 506.3 m	0.172		0.341	0.172
2	506.3	Reflective fault	0.615	-29		0.787
		Sec. 716.2 m	0.179		0.250	0.966
3	1222.5	Non Reflective	0.279			1.245
		Sec. 716 m	0.239		0.334	1.484
4	1938.6	Non Reflective	0.102			1.586
		Sec. 711.6 m	0.242		0.340	1.828
5	2650.1	Reflective fault	---	>-20		1.828
		1550nm				
1	0.0000	Launch Level	---	-49		0.000
		Sec. 506.3 m	0.096		0.190	0.096
2	506.3	Reflective fault	1.194	-32		1.291
		Sec. 1434.3 m	0.886		0.468	2.177
3	1938.0	Non Reflective	0.182			2.358
		Sec. 711.8 m	0.135		0.190	2.493
4	2650.1	Reflective Fault	---	>-18		2.493

From the measured values can be seen that the attenuation of 1310nm is about 0.7dB less. The total length of the route is 2143.8m as the length of series fibre is measured.

The optical unit consists of three optical fibres (approx. 706m) connected with optical welds. At a wavelength of 1550nm the optical weld at 1225m did not occur. Higher attenuation is caused by the optical fibre bend placed after the series fibre. The optical welds have a tendency to occur at the wavelength of 1310nm. In Fig.6 is designed the optical network with PLC splitter and the final values are in TABLE II.

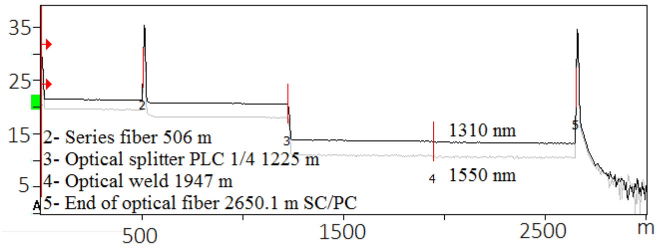


Fig. 6. The backscatter curve OTDR with PLC at 1310nm and 1550nm.

TABLE II. VALUES FOR WITH PLC AT 1310 nm AND 1550 nm

No	Loc. (m)	Event Type	Loss (dB)	Refl. (dB)	Att. (dB/km)	Cum. (dB)
		1310nm				
1	0.0000	Launch Level	---	-43		0.000
		Sec. 506.3 m	0.172		0.340	0.172
2	506.3	Reflective fault	0.546	-31		0.718
		Sec. 719.0 m	0.180		0.250	0.898
3	1225.3	Non Reflective	6.751			7.649
		Sec. 721.8 m	0.290		0.402	7.939
4	1947.1	Non Reflective	0.156			8.095
		Sec. 704.9 m	0.226		0.321	8.322
5	2650.1	Reflective fault	---	-16		8.322
		1550nm				
1	0.0000	Launch Level	---	-42		0.000
		Sec. 506.3 m	0.110		0.216	0.110
2	506.3	Reflective fault	1.319	-33		1.429
		Sec. 720.1 m	0.173		0.240	1.602
3	1226.4	Non Reflective	6.996			8.598
		Sec. 1423.7 m	0.510		0.735	9.108
4	2650.1	Reflective Fault	---	-15		9.108

The attenuation varied from 8dB to 9dB during the measurements. The optical splitter is used at a distance of 719.1m. The attenuation is higher at 1550nm due to bending after a series fibre. More participants can be incorporated into the optical networks by using fibre optic splitter but the attenuation is in the range of approx. 10dB.

V. Conclusion

Two types of optical networks were implemented between the buildings at TUKE. Simulations were completed prior to implementation of the optical networks in order to avoid adverse phenomena. The values in the simulations were higher than the values measured in the implementation. It was i.e. due to the facts that into the simulations were entered catalogue values for the maximum attenuation. The measurement of the optical network PON achieved the attenuation of about 8dB less than network with PLC. Benefit of this type of optical network is a low attenuation caused by welds and bends of the optical fibre. The disadvantage of the network is a connection of two users only. Hence, this is the P2P type of the optical network. The optical network with PLC had higher attenuation caused by the optical splitter which is the drawback of this network. The main advantage is the distribution of the optical fibre to several users. There are a couple of requirements to be considered while deciding which type of the network to choose: number of users, cost and attenuation. Both optical networks were created at a length of 2150m, therefore it is based on customer preferences what type of network to choose for implementation.

Acknowledgment

This work was supported by research grant KEGA no. “023TUKE-4/2017” and VEGA no. “1/0772/17“.

References

[1] Ľ. Mikuš, “Evaluations of the error rate in backbone networks”, In: Elektrorevue, vol. 12, no. 2, 2010, pp. 1-6.

[2] W. Steingartner, D. Radaković, F. Valkošák, Pavol Macko, “Some properties of coalgebras and their rôle in computer science”, In: Journal of Applied Mathematics and Computational Mechanic, vol. 15, no. 4, 2016, pp. 145-156.

[3] V. Skorpil, R. Precechtel, “Training a neural network for a new node element design,“ Przeglad Elektrotechniczny, vol. 89, no. 2B, 2013, pp.187-192.

[4] W. Steingartner, V. Novitzká, “Coalgebras for modelling observable behaviour of programs,” In: Journal of applied mathematics and computational mechanics, vol. 16, no.2, 2017, pp.145-157.

[5] Ľ. Mikuš, “LMS systém Moodle“, E-learn Žilina 2004, ISBN 80-8070-190-3, 2004, pp. 243-248.

[6] T. Ivaniga, P. Ivaniga, “Comparison of the optical amplifiers EDFA and SOA based on the BER and Q-factor in C-band”, In: Advances in Optical Technologies, 2017, pp.1-9. doi:10.1155/2017/9053582.

[7] J. Smiesko, J. Uramova, “Access node dimensioning for IPTV traffic using effective bandwidth,” In: Komunikacie, ISSN: 1335-4205, vol. 14, no. 2, 2012, pp. 11-16.

[8] L. Figuli, J. Smieško, “Recognition and modelling of bursty period of flow,” In: WSEAS Transactions on Communications, vol. 13, 2014, pp. 444-451.

[9] T. Ivaniga, P. Ivaniga, J. Turán, Ľ. Ovseník, “Analysis of possibilities of increasing the spanned distance using EDFA and DRA in DWDM system” COMMUNICATIONS - Scientific Letters of the University of Zilina, 2017, vol. 19, no.3, pp.88-95.

[10] J. Smiesko, Exponential model of token bucket system, In: Komunikacie, vol. 5, no.4, pp.66-70.

[11] P. Ivaniga, T. Ivaniga, “10 Gbps optical line using EDFA for long distance lines“, Przeglad Elektrotechniczny, vol. 93, no. 3, 2017, pp. 193-196, doi:10.15199/48.2017.03.45.

[12] J. Papán, P. Segeč, M. Drozdová, L. Mikuš, M. Moravčík, J. Hrabovský, “The IPFRR mechanism inspired by BIER algorithm,” 2016 International Conference on Emerging eLearning Technologies and Applications (ICETA), 2016, pp. 257-262.

[13] W. Steingartner, V. Novitzká, “Categorical model of structural operational semantics for imperative language,In: Journal of Information and Organizational Sciences, vol. 40, no. 2, 2016, pp. 203-219.

Resource Oriented BDI Architecture for IDS

Ján Perháč
Department of Computers and Informatics
Faculty of Electrical Engineering
and Informatics
Technical University of Košice
Košice, Slovak Republic
Jan.Perhac@tuke.sk

Daniel Mihályi
Department of Computers and Informatics
Faculty of Electrical Engineering
and Informatics
Technical University of Košice
Košice, Slovak Republic
Daniel.Mihalyi@tuke.sk

Lukaš Maťaš
Proxis, spol. s r.o.
Bardejov, Slovak Republic
Lukas.Matas@proxis.sk

***Abstract*—In this work, we propose a resource-oriented architecture of a rational agent for network intrusion detection. We have designed this architecture based on the well-known IRMA architecture. The proposed architecture describes the behavior of a rational agent after detecting a network intrusion. Then it describes the creation of countermeasures to ward off detected threats. Examples are created based on the proposed architecture, describing the implementation of a rational agent detection. We have described these examples by proposed linear BDI logic behavioral formulæ, that have been proven in Gentzen sequent calculus.**

***Keywords*—BDI Logic, IDS, Linear Logic, Resource Oriented Architecture**

I. Introduction

Computer security and especially network security is a very recent research topic. The number of computers around us increases every day, as they become a part of everyday life. This creates a necessity to increase computer security against malicious activities.

Our recent research is dedicated to the Intrusion Detection Systems (IDS), which are one of the ways of making computer network more secure. For that, we have created a real network laboratory environment, which consists of an attacker and a victim connected to the Local Area Network (LAN), and the Internet. Based on our observations, we strive to design a verifiable formal *coalgebraic model for description of IDS's behavior* [1]. With such a model, it will be possible to eliminate an undesirable behavior of IDS, and increase its efficiency. In [2], we have converted network signatures of real network attacks used by IDS to detect network intrusions [3], to coalgebraic signatures. In [4], we have used coalgebraic signatures to model IDS as a coalgebra of polynomial endofunctor over category of infinite stream of packets. Therefore, we have first constructed mentioned category, and then we have specified a polynomial endofunctor over this category. In [5], we have specified the behavior of IDS during attacks as a formula of our modal linear logic, which we have proven in Gentzen sequent calculus. In [6], we have constructed a time-spatial structure of the formula from [5] using Girard's time-spatial theory called Ludics.

One of the future goals of our work is to extend IDS passive reactions at detected intrusions (expects system administrator intervention) to active ones. To do so, we have chosen Belief-Desire-Intention (BDI) logical system. In this paper, we define our linear BDI logic, based on the BDI and linear logic. Based on our logic, we propose new BDI architecture for IDS and we show a motivation example of its behavior during specific intrusion.

II. Basic notions from BDI

A BDI logic controls decisions of BDI architecture agents. A BDI architecture is a philosophical theory of practical reasoning, comming from human reasoning and deduction based on the following attitudes: belief, desire, and intention [7]. Those attitudes represent mental situations that the BDI agent may acquire. The BDI agent is "base unit" of the BDI architecture. It is based on the action chosen by the process of reasoning. This process consists of the following two steps.

- In the first step, we define the database of desires, based on the current belief of the agent.
- In the second step, it will be decided how specific desires will be met by actions from the intention database.

In terms of IDS, it is possible to use the BDI logic as follows: after gaining beliefs about (network) intrusion, it is possible to realize desires through intentions (plans). Principle of our goal is depicted in the Fig. 1.

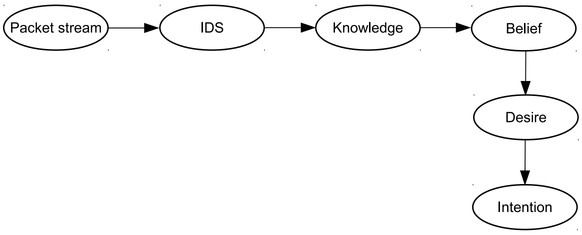


Fig. 1. Usage of BDI logic to secure a computer system

A BDI logic could be used not just as a formal specification language for agents [8], but also as a basic method of implementation of agents in BDI architectures [9]. Specifically, it is used for specification, design and verification of certain behavioral properties of agents.

A BDI logic introduces new modal connectives, that allow a description of a mental situation of an agent [10]. An agent can acquire the following mental situations:

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

- *Belief*. This situation expresses what information an agent possesses in the current state and in the world in which it exists.
- *Desire*. This situation expresses what the agent wants to achieve, i.e. which states will occur.
- *Intention*. This situation expresses what the agent can perform, i.e., the facts that will lead to the desires.

A. Agents

Nowadays, there is no simple or general definition of what an agent is. In many publications there is a plethora of different definitions and divisions of agents. For example, Wooldridge in [11] divides agents to weak and strong based on specific properties that agents possess. Another division of agent systems [12] is to single-agent systems and to multi-agent systems.

Based on the autonomy [13], an agent can be described as an entity that works without direct human or other intervention and possesses control over its internal state. They can be divided by following properties.

- *Intelligent agent* has ability to meet goals in its interest, using his own ”intelligence,” in most cases by a logical deduction.
- *Reactive agent* immediately responds to certain environmental changes without having an internal representation of the environment.
- *Deliberative agent* has ability to plan the progress of its actions leading to the achievement of the chosen goals or desires.
- *Cognitive agent* has the ability to deduct logical conclusions from its observations of the environment. In particular, such an agent must be able to learn and create its own knowledge.
- *Rational agent* has all of the above properties and its structure contains both a planning and a cognitive base including the knowledge base. It is an agent that based on its own knowledge, is able to learn and then plan its activities to achieve its goals (desires).

Agent system is the set of the agents within the environment in which they operate. In our work, we use a single-agent system, with rational agents [13]. Rational agent is able to learn and based on that, plan its next activities in order to achieve its objectives.

B. Architectures

For BDI system, there are several architectures such as: Jadex [14], PRS [15], or IRMA. The IRMA architecture is the first architecture mentioned in the approach based on BDI [16]. It is designed for planning and practical reasoning in the real environment, with limited time for decision-making and knowledge about the environment. The IRMA architecture is depicted in the Fig. 2.

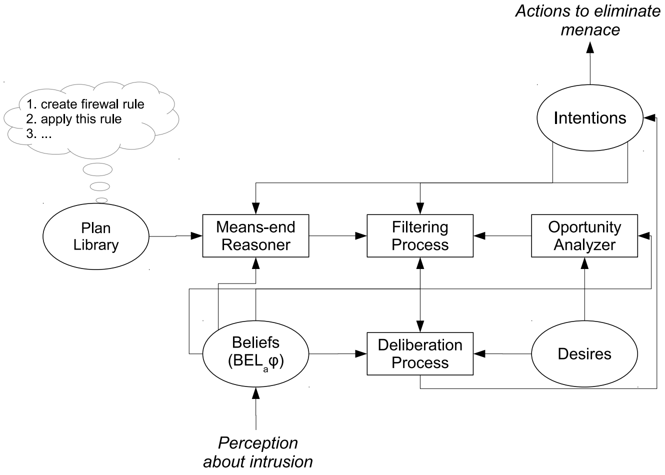


Fig. 2. IRMA architecture

In this architecture, much attention is devoted to the problem of planning. Planning a solution consists of methods of environment examination and sequence of the actions, which will achieve a number of useful ways to create the final plan. Rational agent must consider the final plan and consider alternative practices and appropriate communication during the process. Everything must be done while working with limited resources, and by performing calculations in a specific time.

C. BDI Logic

The original language of the BDI logic is based on the a $\mathcal{L}_{BDI}$ mentioned in [17]. It consists of the following:

- logical connectives of propositional logic:
$$\neg(.), \wedge, \rightarrow, \qquad (1)$$
- and modal connectives, which represent mental situation of an agent:
$$BEL_i, DES_i, INT_i, \qquad (2)$$
where $i \in A$, and i represents the name of the agent, and A is the set of the agent names.

Bratman et all extended this language in their work [10] about temporal logical connectives to the Computation Tree Logic (CTL).

III. Linear BDI Logic

Linear BDI logic is a logic of reasoning over agents based on resource oriented linear logic. In our approach, we define linear BDI logical system as follows:

- we have translated logical connectives of the propositional logic in to the multiplicative fragment of linear logic:
$$(.)^{\perp}, \parr, \otimes, \multimap . \qquad (3)$$
- We have left definition of the BDI modal operators BEL_i, DES_i, and INT_i as it is.
- Extension of temporal connectives syntax is not necessary, because the linear logics theory of Ludics came with the possibility that allows a desription of time and space.

Our definition of linear BDI logic (LBDI) language increases its expressing power. It can describe processes of the real world such as causality, or resource consumption. Also, its syntax became quite simpler, and because of the linear logic time-spatial theory of Ludics, it is not necessary to extend its language by temporal logical connectives.

A. Syntax of LBDI

The syntax of the LBDI logical system can be expressed by following production rule in Backus-Naur form:

$$\begin{aligned}\varphi ::= \; & p \mid \mathbf{1} \mid \bot \mid \varphi^{\perp} \mid \varphi \otimes \psi \mid \varphi \parr \psi \mid \varphi \multimap \psi \\ & \mid BEL_i(\varphi) \mid DES_i(\varphi) \mid INT_i(\varphi),\end{aligned} \tag{4}$$

where:

- p are elementary formulæ,
- $\varphi^{\perp}$ is linear negation, which expresses duality between action (φ) and reaction ($\varphi^{\perp}$).
- $\varphi \multimap \psi$ is linear implication, which expresses that a (re)action ψ is a causal consequence of action φ and after performing this implication, the resource φ became consumed ($\varphi^{\perp}$).
- $\varphi \otimes \psi$ is multiplicative conjunction, which expresses the performing of both actions simultaneously. It has neutral element **1**.
- $\varphi \parr \psi$ is multiplicative disjunction, which expresses commutativity of duality between available and consumed resources by performing either action φ or action ψ. It has neutral element $\bot$.
- $BEL_i(\varphi)$ denotes that the agent i beliefs that φ.
- $DES_i(\varphi)$ denotes that the action φ is desire of the agent i.
- $INT_i(\varphi)$ denotes that the action φ is intention of the agent i.

B. Proof system of LBDI

We define proof system of the LBDI by Gentzen's sequent calculus. One can define proof system through either the right side form or double side form:

$$\underset{\textit{right side sequent}}{\vdash \varphi, \psi\Delta} \qquad \underset{\textit{double sided sequent}}{\Gamma, \varphi \vdash \psi, \Delta} \tag{5}$$

where Γ,Δ are linear contexts for formulæ φ, ψ. We have chosen double sided sequent form. The inference rules for our LBDI are listed bellow.

1) Identity rule is an axiom i.e. is the only rule which has no assumptions. It expresses tautology: from action φ you can prove reaction φ.

$$\frac{}{\varphi \vdash \varphi}{}^{(id)}$$

2) Structural rules are a cut rule and the exchange rules:

$$\frac{\Gamma \vdash \varphi \quad \Delta, \varphi \vdash \psi}{\Gamma, \Delta \vdash \psi}{}^{(cut)}$$

Exchange rules express commutative property of logic by allowing permutation of formulæ on both sides of the sequent.

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \psi, \varphi \vdash \Delta}{}^{(ex_l)} \quad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \psi, \varphi, \Delta}{}^{(ex_r)}$$

3) Logical rules deal with logical connectives:

$$\frac{\Gamma \vdash \Delta}{\Gamma, \mathbf{1} \vdash \Delta}{}^{(\mathbf{1}_l)} \quad \frac{}{\vdash \mathbf{1}}{}^{(\mathbf{1}_r)} \quad \frac{}{\bot \vdash}{}^{(\bot_l)} \quad \frac{}{\Gamma \vdash \bot, \Delta}{}^{(\bot_r)}$$

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \otimes \psi \vdash \Delta}{}^{(\otimes_l)} \quad \frac{\Gamma \vdash \varphi, \Delta \quad \Phi \vdash \psi, \Sigma}{\Gamma, \Phi \vdash \varphi \otimes \psi, \Delta, \Sigma}{}^{(\otimes_r)}$$

$$\frac{\Gamma \vdash \varphi, \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \multimap \psi \vdash \Delta, \Sigma}{}^{(\multimap_l)} \quad \frac{\Gamma, \varphi \vdash \psi, \Delta}{\Gamma \vdash \varphi \multimap \psi, \Delta}{}^{(\multimap_r)}$$

$$\frac{\Gamma, \varphi \vdash \Delta \quad \Phi, \psi \vdash \Sigma}{\Gamma, \Phi, \varphi \parr \psi \vdash \Delta, \Sigma}{}^{(\parr_l)} \quad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \parr \psi, \Delta}{}^{(\parr_r)}$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^{\perp} \vdash \Delta}{}^{(()_l^{\perp})} \quad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \varphi^{\perp}}{}^{(()_r^{\perp})}$$

4) In [18] Nide and Takata introduced the inference rules to deal with modal BDI logical connectives:

$$\frac{\Gamma \vdash \varphi}{BEL_i(\Gamma) \vdash BEL_i(\varphi)}{}^{(BEL_i)}$$

$$\frac{\Gamma \vdash \varphi}{DES_i(\Gamma) \vdash DES_i(\varphi)}{}^{(DES_i)}$$

$$\frac{\Gamma \vdash \varphi}{INT_i(\Gamma) \vdash INT_i(\varphi)}{}^{(INT_i)}$$

IV. Design of Resource Oriented BDI Architecture for IDS

Our resource-oriented BDI architecture for IDS is based on the resource oriented architecture – IRMA. It is designed for planning and practical decision-making of a rational agent after the detection of an intrusion in the computer network. High attention is paid to the problem of planning, where the sequence of actions, which will be performed after detected network intrusion, is chosen.

A. Structure of Resource Oriented BDI Architecture for IDS

Main structures of our architecture are following.

- *Plan library* - are sequences of actions performed after detected network intrusion.
- *Beliefs* - represent the current situation of the agent that is convinced of an intrusion in the computer network. The knowledge about the intrusion, which implies the belief, is acquired by analyzing the network traffic and the following warning about performed intrusion by IDS.
- *Desires* - represent the agents motivation. These are the goals of a rational agent to implement a plan by which an intrusion will be eliminated.

- *Intentions* - represent the situation of the agent, when it is already determined that elimination of a network intrusion is necessary. It is a situation when the agent is convinced, that the proposed plan is appropriate to eliminate the intrusion.
- *IP database* - it is a list of IP addresses from which intrusions were made.
- *Administrator* - it is an e-mail account for alerting the administrator about the intrusion.
- *FW database* - it is a rule database, which is added to the firewall.
- *End of plan* - serves for terminating and disabling access to the host network.

Main aspects of our architecture to achieve rational agent plans are the following.

- *Planning process* - is a process when a rational agent creates a plan by comparing rational beliefs about a realization of an intrusion using the IP address database and the firewall rules.
- *Opportunity analyzer* is a process when a rational agent decides about the believed options, and the options that it wants to include in the plan.
- *Filtering process* is a process in which an agent re-evaluates current plans with propositions to change the plans that were proposed in the Opportunity analyzer.

Design of our Resource-Oriented BDI Architecture for IDS (ROAIDS) is depicted in the Fig. 3.

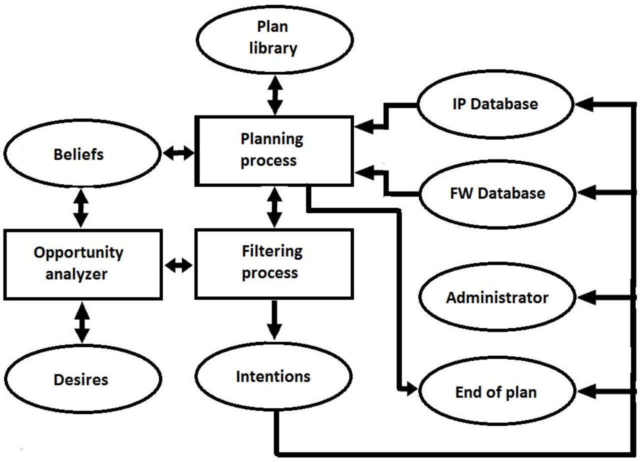


Fig. 3. Resource-oriented BDI architecture for IDS

B. Behavior of ROAIDS during Port Scan Intrusion

In this motivation example, we describe a behavior of our architecture during *Port Scan* intrusion. This intrusion is very popular prerequisite attack technique. It is designed to detect running services and open ports [19]. All computers connected to a LAN or the Internet have many services running and have many ports enabled. Port scanning consists of sending messages to all ports. Then, depending on the type of the response, attacker can determine which port is used and which is not. Based on this information, the attacker can focus their attacks on the ports and services by using known errors.

Key aspect of our architecture is the Plan library, which holds the sequence of actions that are to be performed for identifying and eliminating detected network intrusion.

The Plan library contains a following sequence of steps for a rational agent.

1) *Analysis of network traffic* - In this step the architecture watches for a pattern of known types of network intrusions. If a suspicious pattern is detected, it will log alert about realization of intrusion. Let us denote it as φ.
2) *Addition of IP address into IP database* - This step will add an IP address of the attacker in to the IP database. Let us denote it as ψ.
3) *Creation and addition of firewall rules* - This will create appropriate firewalls rules and add them into the firewall rule database. Let us denote it as γ.
4) *Alert of administrator* - In case of detected Port Scan intrusion. This step will send information about type of intrusion and information about attacker to the administrator. Let us denote it as δ.
5) *End of plan* In the last step, the architecture terminates the execution of a rational agent after intrusion detection and execution of all necessary countermeasures. Let us denote it as ϑ.

The planning process begins to deal with detected intrusion by choosing an appropriate plan, by comparing the beliefs of rational agent $BEL_a\varphi$, obtained from the alert by analysis of network traffic, together with the IP database ψ and the FW database γ.

The situation when the attackers IP address doesn't have match with IP database, and firewall (FW database) does not contain appropriate rules for eliminated alert about Port Scan intrusion $\gamma^{\perp}$, can be expressed by following formula of LBDI logic.

$$BEL_a\varphi \otimes \psi^{\perp} \otimes \gamma^{\perp}. \tag{6}$$

We have proven this formula (6), using GSK defined in the section III-B. Proof of the formula in GSK is a proof tree, where root of this tree is the sequent containing formula. Every step of deduction is created by using an appropriate rule, until all leaves are identities (axioms). The proof is depicted in the Fig. 4.

Rational agent a will send a plan to the Filtering process, where it will be compared by the Opportunity analyzer. Then, during the process of opportunities analyzing, rational agent will compare its belief about the Port Scan $BEL_a\varphi$ to comply with desires of rational agent. In this case, the desires are following.

- Add the IP address of attacker into the IP database: $DES_a\psi$.
- Create appropriate firewall rules and add them into FW database: $DES_a\gamma$.
- Alert administrator about realization of Port Scan: $DES_a\delta$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi,\psi,\gamma\vdash\varphi,\psi,\gamma}\,(\mathrm{id})}{BEL_a\varphi,\psi,\gamma\vdash BEL_a\varphi,\psi,\gamma}\,(\mathrm{BEL_i})}{BEL_a\varphi,\psi^\perp,\gamma,\psi\vdash BEL_a\varphi,\gamma}\,((\,)^\perp_{\mathrm{l}})}{BEL_a\varphi,\psi^\perp,\gamma\vdash BEL_a\varphi,\psi^\perp,\gamma}\,((\,)^\perp_{\mathrm{r}})}{BEL_a\varphi,\psi^\perp,\gamma^\perp,\gamma\vdash BEL_a\varphi,\psi^\perp}\,((\,)^\perp_{\mathrm{l}})}{BEL_a\varphi,\psi^\perp,\gamma^\perp\vdash BEL_a\varphi,\psi^\perp,\gamma^\perp}\,((\,)^\perp_{\mathrm{r}})}{BEL_a\varphi\otimes\psi^\perp,\gamma^\perp\vdash BEL_a\varphi,\psi^\perp,\gamma^\perp}\,(\otimes_{\mathrm{l}})}{BEL_a\varphi\otimes\psi^\perp\otimes\gamma^\perp\vdash BEL_a\varphi,\psi^\perp,\gamma^\perp}\,(\otimes_{\mathrm{l}})$$

Fig. 4. Proof of the formula (6)

- End plan: $DES_a\vartheta$.

Mentioned process can be described by the following formula of LBDI logic.

$$BEL_a\varphi\otimes DES_a\psi\otimes DES_a\gamma\otimes DES_a\delta\otimes DES_a\vartheta. \tag{7}$$

After comparing rational agents beliefs and desires by the Opportunity analyzer, the agent will send plan to the Filtering process. Based on the results from planning process and Opportunity analyzer, the plan will be reevaluated. This process can be expressed by the following formula of LBDI logic.

$$\begin{gathered}(BEL_a\varphi\otimes\psi^\perp\otimes\gamma^\perp)\otimes\\(BEL_a\varphi\otimes DES_a\psi\otimes DES_a\gamma\otimes DES_a\delta\otimes DES_a\vartheta).\end{gathered}\tag{8}$$

After that, the rational agent will send the plan to the *Intentions* structure, where the following intentions will be created.

- Add the IP address of an attacker into the IP database: $INT_a\psi$.
- Add firewall rules into the FW database: $INT_a\gamma$.
- Alert administrator: $INT_a\delta$.
- End plan: $INT_a\vartheta$.

The process of creation of desires to eliminate network intrusion can be expressed by following formula of the LBDI logic.

$$\begin{gathered}((BEL_a\varphi\otimes\psi^\perp\otimes\gamma^\perp)\\\otimes\\(BEL_a\varphi\otimes DES_a\psi\otimes DES_a\gamma\otimes DES_a\delta\otimes DES_a\vartheta)))\\\multimap\\(INT_a\psi\otimes INT_a\gamma\otimes INT_a\delta\otimes INT_a\vartheta).\end{gathered}\tag{9}$$

A plan is realized according to the created desires as follows.

- Add the IP address of an attacker into the IP database: ψ.
- Add firewall rules into the FW database: γ.
- Alert administrator: δ.
- End plan: ϑ.

The whole process of realization of a plan for identification and elimination of the network intrusion can be expressed by following formula of the LBDI logic.

$$\begin{gathered}(((BEL_a\varphi\otimes\psi^\perp\otimes\gamma^\perp)\otimes\\(BEL_a\varphi\otimes DES_a\psi\otimes DES_a\gamma\otimes DES_a\delta\otimes DES_a\vartheta))\multimap\\(INT_a\psi\otimes INT_a\gamma\otimes INT_a\delta\otimes INT_a\vartheta))\multimap\\(\psi\otimes\gamma\otimes\delta\otimes\vartheta)\end{gathered}\tag{10}$$

The behavior of our architecure during a situation when attacker tries to perform port scan from the same or a different IP address is as follows.

- *Plan library* has the same sequence of actions as in the case of first Port Scan.
- *Planning process*. Agent shall process its rational belief about the intrusion $BEL_a\varphi$.
    - ○ Agent will compare attacker's IP address to IP database ψ. In case that the match is not found,
    - ○ agent will check if the FW database contains an appropriate firewall rules γ.

Behavior of the architecture after a repeated port scan network intrusion can be expressed by the following formula of the LBDI logic.

$$(BEL_a\varphi\otimes(\psi\parr\gamma))\multimap\vartheta \tag{11}$$

The proof of the formula (11) is depicted in the Fig. 5.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi\vdash\varphi}\,(\mathrm{id})}{BEL_a\varphi\vdash BEL_a\varphi}\,(\mathrm{BEL_i})}{\vdash BEL_a\varphi,BEL_a\varphi^\perp}\,((\,)^\perp_{\mathrm{r}})\qquad\dfrac{\dfrac{\dfrac{\dfrac{}{\psi,\gamma\vdash\psi,\gamma}\,(\mathrm{id})}{\psi\vdash\psi,\gamma,\gamma^\perp}\,((\,)^\perp_{\mathrm{r}})}{\vdash\psi,\gamma,\psi^\perp,\gamma^\perp}\,((\,)^\perp_{\mathrm{r}})}{\vdash\psi\parr\gamma,\psi^\perp,\gamma^\perp}\,(\parr_{\mathrm{r}})}{\vdash BEL_a\varphi\otimes(\psi\parr\gamma),BEL_a\varphi^\perp,\psi^\perp,\gamma^\perp}\,(\otimes_{\mathrm{r}})\qquad\dfrac{}{\vartheta\vdash\vartheta}\,(\mathrm{id})}{(BEL_a\varphi\otimes(\psi\parr\gamma))\multimap\vartheta\vdash BEL_a\varphi^\perp,\psi^\perp,\gamma^\perp,\vartheta}\,(\multimap_{\mathrm{l}})$$

Fig. 5. Proof of the formula (11)

V. Conclusion

In this paper, we propose resource oriented BDI architecture for IDS. Because of that, we have created its controlling logical system based on the BDI logic and linear logic. We have simulated a Port Scan intrusion and described behavior of this architecture step by step. Every step is described by a provable formula of our LBDI logic. Here, we present only the first and last steps proofs in Gentzen sequent calculus. Design of our architecture allows automated response of IDS to incoming intrusions.

In the future, we would like to extend our approach with implementation of our architecture into a real network environment. We plan to combine a network intrusion detection system with a host-based intrusion detection system. This would increase the security of computer systems even more.

ACKNOWLEDGMENT

This paper was supported by KEGA project ViLMA: Virtual Laboratory for Malware Analysis (079TUKE-04/2017).

This work is a result of international cooperation under the CEEPUS network No.CIII-HU-0019-12-1617.

REFERENCES

[1] J. Perháč, D. Mihályi, and V. Novitzká, “Design of verifiable model of program systems’ complex security using coalgebras and coalgebraic logics,” in *Electrical Engineering and Informatics 7 : proceedings of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice*, Košice, FEEI TU, 2016, pp. 120–124.

[2] J. Perháč and D. Mihályi, “Coalgebraic specification of network intrusion signatures,” *Studia Universitatis Babes-Bolyai, Informatica*, vol. 61, no. 2, pp. 83–94, 2016.

[3] M. Roesch, “Snort users manual,” https://www.snort.org/, citované: 18.11.2016.

[4] J. Perháč and D. Mihályi, “Coalgebraic modeling of ids behavior,” in *2015 IEEE 13th International Scientific Conference on Informatics*, November 18-20, 2015, Poprad, Slovakia, Danvers: IEEE, 2015, pp. 201–205.

[5] ——, “Intrusion detection system behavior as resource-oriented formula,” *Acta Electrotechnica et Informatica*, vol. 15, no. 3, pp. 9–13, 2015.

[6] J. Perháč, D. Mihályi, and V. Novitzká, “Between syntax and semantics of resource oriented logic for ids behavior description,” *Journal of Applied Mathematics and Computational Mechanics*, vol. 15, no. 2, pp. 105–118, 2016. [Online]. Available: http://dx.doi.org/10.17512/jamcm.2016.2.12

[7] M. J. Wooldridge, *Reasoning about rational agents*. MIT press, 2000.

[8] E. A. Emerson and J. Srinivasan, “Branching time temporal logic,” in *Workshop/School/Symposium of the REX Project (Research and Education in Concurrent Systems)*. Springer, 1988, pp. 123–172.

[9] M. P. Singh, A. S. Rao, and M. P. Georgeff, “Formal methods in dai: Logic-based representation and reasoning,” in *Multiagent systems*. MIT Press, 1999, pp. 331–376.

[10] M. E. Bratman, D. J. Israel, and M. E. Pollack, “Plans and resource-bounded practical reasoning,” *Computational intelligence*, vol. 4, no. 3, pp. 349–355, 1988.

[11] M. Wooldridge and N. R. Jennings, “Intelligent agents: Theory and practice,” *The knowledge engineering review*, vol. 10, no. 2, pp. 115–152, 1995.

[12] ——, “Agent theories, architectures, and languages: a survey,” in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1994, pp. 1–39.

[13] A. Kubík, *Inteligentni agenty: tvorba aplikacniho software na bazi multiagentovych systemu*. Computer Press, 2004.

[14] A. Pokahr, L. Braubach, and W. Lamersdorf, “Jadex: A bdi reasoning engine,” *Multi-agent programming*, pp. 149–174, 2005.

[15] M. P. Georgeff and A. L. Lansky, “Reactive reasoning and planning.” in *AAAI*, vol. 87, 1987, pp. 677–682.

[16] A. Kubík, *Agenty a multiagentové systémy*. Slezská univerzita, Filozoficko-přírodovědecká fakulta, Ústav informatiky, 2000.

[17] V. Nair, *On Extending BDI Logics*. Griffith University, 2003.

[18] N. Naoyuki and S. Takata, “Deduction systems for bdi logics using sequent calculus,” in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*. ACM, 2002, pp. 928–935.

[19] M. de Vivo, E. Carrasco, G. Isern, and G. O. de Vivo, “A review of port scanning techniques,” *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 2, pp. 41–48, Apr. 1999. [Online]. Available: http://doi.acm.org/10.1145/505733.505737

MEMS technology in optical switching

1st Ivan Plander
European Polytechnic Institute
Kunovice, Czech Republic
ivan.plander@tnuni.sk

2nd Michal Stepanovsky
Faculty of Information Technology
Czech Technical University in Prague
Praha 6, Czech Republic
michal.stepanovsky@fit.cvut.cz

***Abstract*—All-optical switching fabrics based on the Micro-Electro-Mechanical Systems (MEMS) technology are now widely available on the market. This paper reviews working principles and architectures of MEMS-based optical switches from the past to the present day. During the last two decades, many approaches and actuating mechanisms emerged. In general, optical MEMS-based switches can employ movable micromirrors as basic switching element, or movable couplers (resonators). The switching speed can vary from less than 1μs to tens of ms. This paper briefly describes principles of all of these switches. Optical properties (insertion loss, crosstalk) and switching speed are also discussed.**

***Index Terms*—Optical Networks, Optical switching, MEMS technology, Free-space switches, Waveguide switches, State-of-the-art**

I. Introduction

All-optical switching has emerged as the alternative to electronic switching in realizing all-optical networks. There are three different switching techniques for all-optical networks: the optical circuit switching (OCS), optical burst switching (OBS) and optical packet switching (OPS). Optical burst switching is a technology positioned between circuit switching and packet switching. Both, OCS and OBS does not require optical buffering or packet-level parsing and the optical datapath is reserved in advance. MEMS technology provides significant advantages (low crosstalk, polarization and wavelength insensitivity, good scalability, etc.) in realizing OCS and OBS networks. $N \times N$ switching fabric allow for simultaneous connection between N input fibers and N output fibers, in a fully non-blocking all-optical manner, as it is shown in Fig. 1.

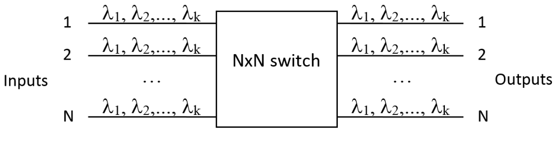


Fig. 1. Simplified architecture of the all-optical switching node

There are two major types of optical switches on the market[1]: **opto-mechanical** optical switches and **MEMS-based** optical switches.

a) Opto-mechanical optical switch: is an old type of switches, but still used. It relies on the movement of optical fiber or optical element to direct the optical path from input fiber to output fiber, e.g. it can utilize a high resolution stepper motor to directly connect fiber to one another, or to move a prism between fixed collimators. The advantage is a low insertion loss and low crosstalk, however the main disadvantage is slow switching speed. Thus, it is only applicable for large data steady flows. Opto-mechanical optical switches have different configurations such as 1×2, 1×4, 1×8, 1×16, etc.

b) MEMS-based optical switch: usually uses tiny micromirrors of diameter of the order of some hundreds of micrometers to reflect a optical beam inside the switch. In that way, the input optical signal can be directed to any out port. Comparing to opto-mechanical switches, these type of switches provide significantly faster switching speed while still providing low insertion loss, low crosstalk and wide wavelength range. Typical configurations of MEMS-based switches are 1×2,..., 1×16, 2×2,..., 32×32, or even 128×128, 256×256 or 512×512. Recently developed industrial all-optical switches are usually based on MEMS technology and many developers and companies offer their solutions. These optical switches have found use in communication networks that transfer large bandwidth data over long distances. Another type of MEMS-based switches uses a coupler/resonator (microring, microdisc or a directional coupler) to control the optical signal inside the switch instead of using micromirrors. This coupler/resonator is typically tuned for a specific wavelength, so the it can be used for the Wavelength Selective Switching (WSS). However, recent demonstrations show that optical switches based on adiabatic coupler provide both fast switching speed (less than 1 μs) and wide optical bandwidth (about 300 nm).

Today's MEMS-based optical switches on the market are developed and offered by various companies, e.g. Glimmerglass (www.glimmerglass.com), Calient Technologies (www.calient.net), DiCon Fiberoptics (www.diconfiberoptics.com), IntelliSense (www.intellisense.com), Fujitsu (www.fujitsu.com), Sercalo Microtechnology (www.sercalo.com), Lumentum (www.lumentum.com) and others.

II. Classification of MEMS-based switches

According to the optical beam propagation inside the switch, MEMS-based switches can be divided into two categories – **free-space** switches and **waveguide** switches. In the free-space optical switch the optical signal is propagating in free space

[1] Switches involving mechanical parts during the switching process

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

and along its path is directed to the desired output fiber. In the waveguide optical switch the optical signal is propagating in waveguides. The switching to another waveguide is realized by placing another waveguide (or coupler) to the specific position in that way they are coupled and the optical signal from one waveguide completely transfers to the other waveguide.

According to the optical system configurations, MEMS-based switches can be roughly divided into two categories – **2D MEMS** and **3D MEMS** switches. In a 2D MEMS switch all the optical components are placed on a 2D plane, whereas in a 3D MEMS switch optical components are arranged in a 3D configuration.

III. FREE-SPACE MEMS-BASED SWITCHES

In this section we describe the main types of free-space switch configurations developed in the past and some of them still used in the present day. However, before introducing developed configurations, we will first shortly discuss optical beam propagation in free-space. The optical beam in free-space can be described by a Gaussian beam. The electrical field of a Gaussian beam is a function of its axial coordinate z and its radial coordinate r. The beam radius (defined as the radius at which the electric field amplitude fall to $1/e$ of their axial value) is given by

$$w(z) = w_0\sqrt{1 + (z/z_R)^2}, \tag{1}$$

where $w_0 = w(0)$ is the waist radius (smallest radius of the beam), $z_R = n\pi w_0^2/\lambda$ is Rayleigh distance, n is the refractive index and λ is is the wavelength of signal. Rayleigh distance is the length from the waist along the beam path where its radius is increased by a factor of $\sqrt{2}$.

When the distance between the transmitting plane and the receiving plane is designed to be equal $2z_R$ and the beam first converges to a waist at the midpoint and then diverges (as it is shown in Fig.2), then this beam is the narrowest that may propagate between the two lenses (between the transmitting plane and the receiving plane) [1]. The minimum size of micromirror used to reflect the optical beam along the optical path is determined by the incident beam radius $w(z)$. For micromirror radius R to be equal $w(z)$, the reflection efficiency is limited to 0.865, corresponding to a loss of 0.63 dB. For $R = \sqrt{2}w(z)$, the loss falls to 0.08 dB [1].

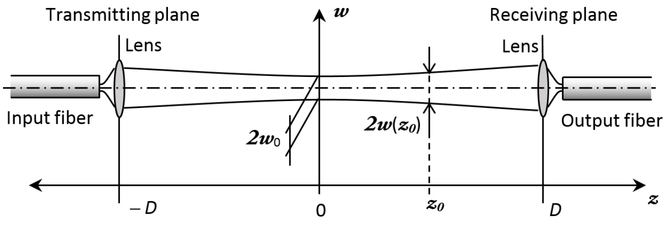


Fig. 2. Gaussian beam propagation inside the switch.

A. 2x2 vertical torsion mirror switch

This type of switch consists of four vertical torsion micormirrors [2]. The principle of the switch operation is shown in Fig.3 and Fig.4. The vertical mirrors are arranged such that in the through mode, the mirrors are rotated out of the optical paths and the input beams directly propagate into the opposite output fibers. In the reflection mode, the input beams are reflected to the output fibers on the same side of the chip. The design reported in 1998 uses electrostatic actuation with the switching time less than 1 ms [2].

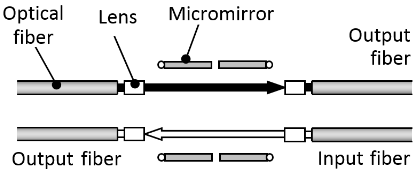


Fig. 3. 2x2 vertical torsion mirror switch – THROUGH mode

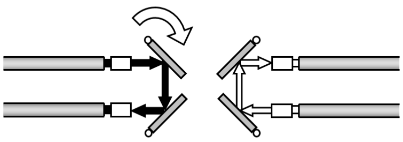


Fig. 4. 2x2 vertical torsion mirror switch – REFLECTION mode

B. 2x2 switch with single double-sided mirror

The mirror is attached to a long beam that is electrostatically actuated by comb drives. Depending on the actuator design the switch, the micromirror can have one or two stable positions. The first design from 1998 achieves the switching time below 1 ms and insertion loss in reflection mode about 1.6 dB [3]. The improved design from 1999 achieves the switching time about $500\mu s$,the insertion loss is between 0.3 and 0.5 dB and the cross-talk about -70 dB [4]. This type of configuration is still available in the market. For instance, the optical switch offered by Sercalo in late 2017 have similar parameters (0.5 ms of switching time, 0.4 insertion loss, cross-talk -75 dB) [5] as the improved design from 1999.

Comparing to the design described in the previous section – Section III-A, this configuration uses only single micromirror and comb-drive actuator, which allows for faster switching.

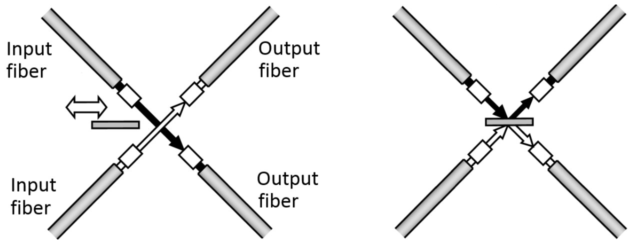


Fig. 5. 2x2 switch with single mirror – THROUGH mode (left) and REFLECTION mode (right)

C. 1x4 switch with translating mirror

The design reported in 2004 in [6] uses three electrostatic comb actuators to drive the mirrors linearly into the optical path – see Fig. 6. Both the switch-on and switch-off times are 0.8 ms [6]. In this case, one-sided mirrors are required.

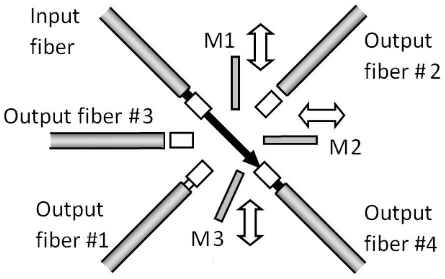


Fig. 6. Principle of 1x4 switch with translating mirrors

D. 1xN switch with rotating mirror

This type of switch consists of one input fiber, N output fibers, central lens and one central micromirror, as it is shown in Fig. 7. The micromirror is mounted on angular actuator, and thus it can reflect incoming optical beam to arbitrary output fiber. The design reported in 2000 have been made using angular micromachined actuator capable of more than ±6° of mechanical motion, the switching time less than than 5 ms and the insertion loss less than 0.5 dB [7].

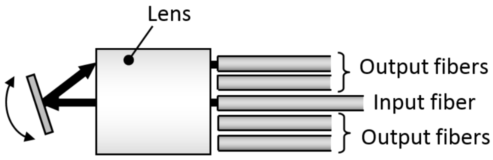


Fig. 7. Principle of 1xN switch with rotating mirror

E. 1xN switch with translating mirror

This switch has one input fiber, N output fibers and N micromirrors each of which belongs to a single output fiber, as it is shown in Fig. 8. The optical beam leaving the input fiber propagates directly until it is reflected by an extended mirror, and thus, the reflected beam is focused onto the desired output fiber. The switch presented in 2000 achieves the switching time of $700\mu s$ and insertion loss about 0.5 dB and uses linear electrostatic comb-driven actuator[7]. The number of output ports do not scale well as the optical path length significantly varies (this increases the optical loss inducted by light beam divergence). However, this switch provides significantly faster switching time and ON/OFF type of micromirrors – comparing to the switch described in previous Section III-D.

F. NxN 2D MEMS switch

All previous designs have their optical components placed on a 2D plane, however 2D MEMS-based switch uses 2D matrix of micromirrors – as it is illustrated in Fig. 9. Actually, micromirrors are arranged in a N×N crossbar architecture, which has the advantages of strict non-blocking routing and simple control algorithm. Micromirrors have only two states ON and OFF. In the ON state, the micromirror is pulled up (down), so it reflects the optical beam from input port to desired output port. The N×N switch requires N^2 mirrors. Thus, the number of mirrors increases considerably with N. The distance between mirrors is proportional to the mirror radius, which increases the switch area significantly with increasing number of ports. Hence, it only allows usually up to 32×32 switch fabrics. Another drawback is the large loss variation between the shortest optical path and the longest one due to Gaussian beam divergence.

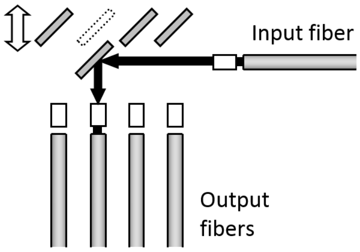


Fig. 8. Principle of 1xN switch with translating mirrors

Optical switch demonstrated in 1998 provided the switching times less than $700\mu s$, crosstalk less than $-60dB$, however significant loss (varying from 14.6 dB for the shortest path to 19.9 dB for the longest path) due to the integrated binary-amplitude Fresnel lenses used only for the demonstration purposes [8]. OMM, originally called Optical Micro-Machines, has been successful in the design and manufacture of the commercial 2D MEMS-based optical switches. Their switch provided the typical switching time about 7 ms, cross-talk less than 50dB, and the insertion loss of 1.7 dB and 3.1 dB for 8×8 and 16×16 switch, respectively [9], [10]. In 2012, CrossFiber Inc. has acquired OMM – substantially all the assets of 2D MEMS switch [11], [12], [13].

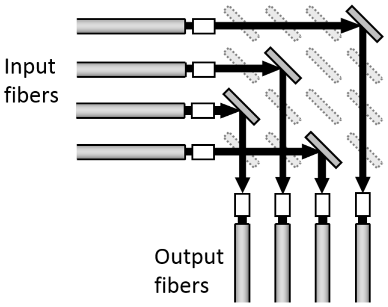


Fig. 9. Principle of 2D MEMS switch

Strictly non-blocking 32x32 switch can be built using four 16x16 switches with an additional passthrough port. The total insertion loss is acceptable since the each switched path goes through only 2 16x16 switches: one where it is reflected by a mirror and one where it just passes through.

To build large-port-count switches, multistage rearrangeable non-blocking 2D MEMS with Benes architecture have been proposed [14]. In this case, all micromirrors are placed on a common chip. It requires $2\log_2 N - 1$ stages and in each stage $N/2$ movable ON/OFF micromirrors and $\frac{N}{2}\log_2 N + \frac{N}{2} - 2$ fixed micromirrors. Another approach proposed in [15] also uses double-sided micromirrors, however in this manner N^2 micromirrors on one substrate can realize two rearrangeable non-blocking switches.

G. NxN 3D MEMS switch

3D MEMS-based switches use micromirrors of diameter of the order of some hundreds of micrometers for optical beam switching between input and output ports. Each micromirror can rotate about two orthogonal axes and its actuation is typically carried out by electrostatic field. The 3D architecture typically employs two arrays of micromirrors, each aligned to an array of collimated input or output fibers. This requires the use of $2N$ mirrors for N ports, considerably less than 2D architecture. The switch consists of an input fiber array, an input lens array, two parallel MEMS mirror arrays in 3D space, an output lens array, and an output fiber array – as it is illustrated in Fig. 10. Each input fiber directs optical beam to a mirror on the input array while the input mirror steers the optical beam to any output mirror, which, in turn, steers the optical beam to an output fiber. Due to the symmetrical design, both input and output mirrors require the same deflection capacity. Uniform lens arrays with optimized focal length are used to collimate the beams in and out of the arrays of fibers. The Fourier transform lens placed between the micromirror arrays allows for several advantages: lower maximum angle requirement, smaller micromirror size, greater tolerance to micromirror curvature, and lower switch crosstalk [16], [17]. Other configurations and improvements of the switch structure are also possible, e.g. [18].

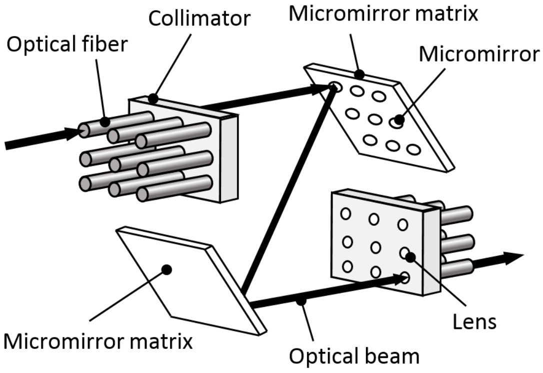


Fig. 10. Principle of 3D MEMS switch

The first commercial deployment of a 3D MEMS based switch was Lucent's 256x256 WaveStar LambdaRouter. The technology was demonstrated at Telecom '99 show in Geneva [19], [20]. The switch offered a few miliseconds of switching time [21], [22]. In 2003, a optical switch with port count exceeding 1100×1100 with mean insertion loss of 2.1 dB and maximum insertion loss of 4.0 dB was demonstrated [16]. Even in 2001, 1296-port switch with a mean insertion loss of 5.1 dB and switching time of 5 ms was reported [23]. The research in this area continued for more than one decade and many improvements were made.

In the recent market (September 2017), there are many products employing 3D micromirrors, e.g. S320 Optical Circuit Switch from Calient Technologies offering maximum insertion loss of 3.5 dB (O,S,C Bands), switching time less than 50 ms and with switching cycles about 10^{12} [24]; or Intelligent Optical System 600 from Glimmerglass offering maximum insertion loss of 3.7 dB, switching time less than 20 ms and the port count from 32×32 to 192×192 [25]; or 32×32 optical switch from DiCon Fiberoptics with maximum insertion loss of 1.5 dB, switching time less than 50 ms and minimum durability of 10^9 cycles [26].

However, as we have shown in [27], [28], [29], [30], the switching time of 3D MEMS optical switch can be significantly reduced if the dynamics of the micromirror is optimized in the way that the rotation of the micromirror about both axes is decoupled. This is realized by matching micromirror eigenfrequencies for its rotation. Thus, the micromirror dynamics can be improved for both open-loop control [27], [28], [29] and closed-loop control [28], [29], [30]. The complex multiphysics model as we have presented in [31], [32] is able to analyze micromirror behavior in detail. Nevertheless, it seems to be more convenient to use hybrid models as presented e.g. in [30], as they provide fast simulation time, and thus, the parameters of the micromirror control unit can be easily optimized.

H. Free-space wavelength-selective MEMS switch

The principle of the wavelength selective switch with a MEMS-micromirror architecture is shown in Fig. 11. The switch can connect any wavelength on the input fiber to any output fibers. For simplicity, it is shown 1x4 switch with four wavelengths (micromirrors). The optical signal enter the switch and is collimated onto a dispersive element, which demultiplexes optical beam into its constituent wavelengths. Fig. 11 shows the dispersive element as reflective grating, however a transmissive diffraction grating can be also used. The individual-wavelength beams leave the grating and is projected onto a dedicated micromirror. The micromirror direct the optical beam to desired output fiber. Usually, a lens is placed between the grating and micormirror array, the optical signal goes thorough grating twice, and optical signal is manipulated in 3D space (not shown in the Fig. 11 for its simplicity). The 1×4 design reported in 2004 in [33] achieves a switching time of less than 380 μs, the later design form 2006 introduces the switch with a switching time less than 0.5 ms and 32 output ports[34].

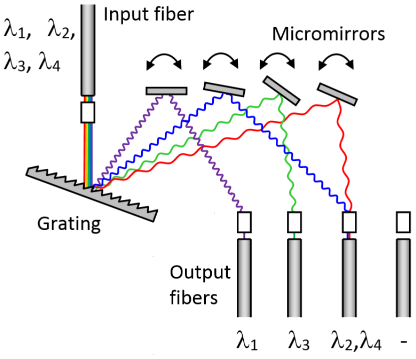


Fig. 11. Principle of WSS switch

IV. Waveguide MEMS-based switches

A. Micro-ring (micro-disc) waveguide MEMS switch

Microdisk or microring resonators offer another principle in optical MEMS-based switching. The ability to physically change the spacing between the waveguide and the microresonator enables to control the coupling coefficient [35]. Fig. 12 shows a principle of a microdisk switch. The coupling coefficient between the microdisk and the waveguide varies exponentially with the gap spacing [36]. The spacing between the microdisk and the waveguide is controlled by pulling the waveguide toward the microdisk by electrostatic force. The switch structure with the vertically coupled waveguides to a resonator is also possible [37].

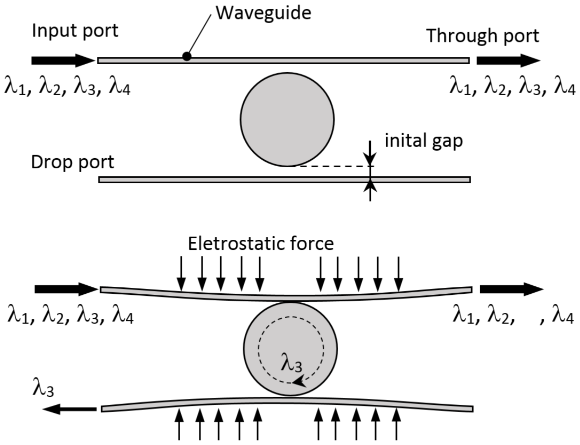


Fig. 12. Principle of microdisc switch with horizontally coupled waveguides

An alternative design uses microring resonator and thin optically lossy bridge placed above the microring [38], [39] – as it is shown in Fig. 13. When the bridge is in contact with the resonator, the quality factor (Q) is significantly lowered, and the resonant wavelength is no longer switched to the drop port. The optical loss introduced by the bridge is a combination of absorption and scattering loss. When the bridge is up and away from the evanescent field, the ring couples the resonant wavelength from the through port to the drop port. As reported in [38] the initial gap between ring and bridge is 2.2μm, the coupling gap between the ring and the waveguides is 185 nm, the switching time is 60μs and can be lowered to 10μs.

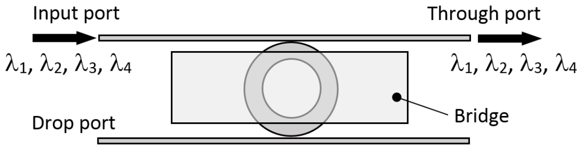


Fig. 13. Principle of microring switch with vertically coupled bridge

B. NxN waveguide MEMS switch

In 2014, 50×50 waveguide switch with the switching time of 3.8 μs has been demonstrated [40], [41]. The switch uses a N×M crossbar architecture, where a pair of directional couplers connected through 90-degree bend is located near to each crossing of input and output waveguide, as it is illustrated in Fig. 14 (M=3, N=3). In the OFF state, the pair of couplers is placed far enough from the input/output waveguides to prevent having optical coupling. In the ON state, the couplers come sufficiently close to the waveguides allowing the transfer of optical power through. In this way, one can switch optical signal from any n-th port to any m-th port.

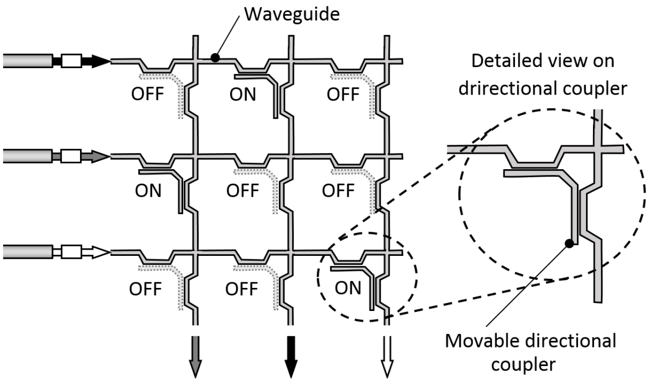


Fig. 14. Digital crossbar switch based on movable couplers

The design presented in [40] uses vertically moving couplers with only two states – ON, OFF. However, the precise control of the coupling brings the advantage of distributing optical power from one port to many ports at the same time. This idea was presented in [41] and uses laterally moving directional coupler, as it is shown in Fig. 15. This coupler allows for sending the optical power only to the desired output ports, what is in contrast to broadcast-and-select approach, which sends optical power to all output ports. In this case, the switching time is 9.6 μs, actuation voltage is 10 V, and the 20 dB bandwidth of the through-port is measured to be 31.5 nm [41].

Recently, 64×64 waveguide switch with vertical adiabatic couplers and with the switching time of 0.91 μs and insertion loss of 3.7 dB has been demonstrated [41], [42], [43]. The adiabatic coupler operates over a broad optical bandwidth (1400–1700 nm), making it fully compatible with wavelength-division multiplexing networks [42]. The adiabatic coupler is

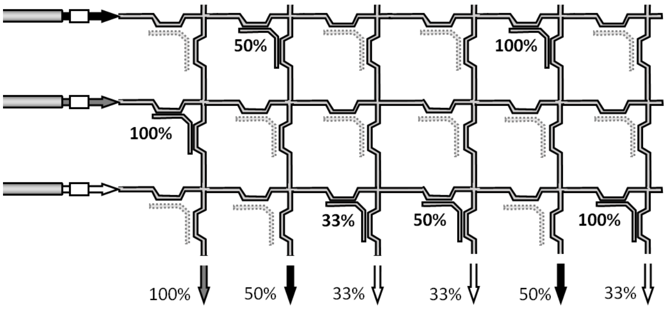


Fig. 15. Analog crossbar switch based on movable couplers

based on proper choice of coupling coefficient and propagation coefficient that ensure adiabatic (slow) evolution of a single normal mode (or eigenmode) of the system, and is tolerant to small to moderate variations of the system's parameters [44]. This switch uses the similar configuration as it is shown in Fig. 14. The matrix of 64×64 vertical adiabatic couplers has been integrated on a 8.6 mm × 8.6 mm chip [42].

V. Conclusion

In this paper, the MEMS-based optical switches was reviewed, including current approaches that are under development. Today's MEMS-based switches available on the market use a micro-mirror to reflect a optical beam inside the switch. The switching speed of these switches is limited by the dynamics of the micromirror, which has to be large enough to reflect the majority of the Gaussian optical beam. Therefore, the switching time bellow one millisecond for ON/OFF type of micromirrors and the switching time of few milliseconds for 3D micromirrors is a typical limit. Recent progress in MEMS-based waveguide optical switches shows that even switching time less than 1 μs is possible to reach. MEMS-based waveguide switches can also provide low insertion loss, good scalability, low cross-talks and wide optical bandwidth. Thus, MEMS-based waveguide optical switches with adiabatic couplers could become a next generation of optical switches replacing conventional 3D MEMS technology.

Acknowledgment

The research was supported by Czech Ministry of Education under Project RVO18000.

References

[1] R. R. A. Syms, "Scaling laws for MEMS mirror-rotation optical cross connect switches," *Journal of Lightwave Technology*, vol. 20, pp. 1084–1094, Jul 2002.

[2] L.-S. Huang, S. sheng Lee, M. E. Motamedi, M. C. Wu, and C.-J. Kim, "Optical coupling analysis and vibration characterization for packaging of 2x2 mems vertical torsion mirror switches," *Proc.SPIE*, vol. 3513, pp. 135–143, 1998.

[3] C. Marxer and N. F. de Rooij, "Micro-opto-mechanical 2x2 switch for single-mode fibers based on plasma-etched silicon mirror and electrostatic actuation," *Journal of Lightwave Technology*, vol. 17, pp. 2–6, Jan 1999.

[4] W. Noell, P. A. Clerc, L. Dellmann, B. Guldimann, H. P. Herzig, O. Manzardo, C. R. Marxer, K. J. Weible, R. Dandliker, and N. de Rooij, "Applications of soi-based optical mems," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 8, pp. 148–154, Jan 2002.

[5] "Miniature fiber optic MEMS switch." \verb=http://se055rz5.edis-lps.ch/products/pdfs/SX2x2-SMF.pdf=, 2017. [Sercalo microtechnology ltd; Datasheet; Online; accessed 19-August-2017].

[6] Z. Wang, W. Cao, X. Shan, J. Xu, S. Lim, W. Noell, and N. de Rooij, "Development of 1x4 mems-based optical switch," *Sensors and Actuators A: Physical*, vol. 114, no. 1, pp. 80 – 87, 2004.

[7] J. D. Grade and H. Jerman, "MEMS electrostatic actuators for optical switching applications," in *OFC 2001. Optical Fiber Communication Conference and Exhibit. Technical Digest Postconference Edition (IEEE Cat. 01CH37171)*, vol. 3, pp. WX2–WX2, March 2001.

[8] L. Y. Lin, E. L. Goldstein, and R. W. Tkach, "Free-space micromachined optical switches with submillisecond switching time for large-scale optical crossconnects," *IEEE Photonics Technology Letters*, vol. 10, pp. 525–527, April 1998.

[9] P. M. D. Dobbelaere, S. Gloeckner, S. K. Patra, L. Fan, C. King, and K. Falta, "Design, manufacture, and reliabilty of 2d mems optical switches," *Proc.SPIE, MEMS/MOEMS: Advances in Photonic Communications, Sensing, Metrology, Packaging and Assembly*, vol. 4945, pp. 39–45, 2003.

[10] P. D. Dobbelaere, K. Falta, S. Gloeckner, and S. Patra, "Digital mems for optical switching," *IEEE Communications Magazine*, vol. 40, pp. 88–95, Mar 2002.

[11] "CrossFiber Acquires IP from Photonic Switch Maker." www.photonics.com/Article.aspx?AID=51985=, October 2012. [Online; Article number: AID=51985; accessed 19-August-2017].

[12] "Optical MEMS switch company CrossFiber raises $13.4 million, acquires OMM." http://www.memsjournal.com/2012/09/optical-mems-switch-company-crossfiber-raises-134-million-acquires-omm.html, September 2012. [Online; accessed 19-August-2017].

[13] "CrossFiber buys assets of MEMS photonic switch maker OMM." http://electroiq.com/blog/2012/09/crossfiber-buys-assets-of-mems-photonic-switch-maker-omm/, September 2012. [Online; accessed 19-August-2017].

[14] X. Ma and G.-S. Kuo, "A novel integrated multistage optical mems-mirror switch architecture design with shuffle benes inter-stage connecting principle," *Optics Communications*, vol. 242, no. 1, pp. 179 – 189, 2004.

[15] G. Shen, T. H. Cheng, C. Lu, T. Y. Chai, and S. K. Bose, "A novel rearrangeable non-blocking architecture for 2d mems optical space switches," *Optical Networks Magazine*, vol. 3, pp. 70–79, Nov 2002.

[16] J. Kim, C. J. Nuzman, B. Kumar, D. F. Lieuwen, J. S. Kraus, A. Weiss, C. P. Lichtenwalner, A. R. Papazian, R. E. Frahm, N. R. Basavanhally, D. A. Ramsey, V. A. Aksyuk, F. Pardo, M. E. Simon, V. Lifton, H. B. Chan, M. Haueis, A. Gasparyan, H. R. Shea, S. Arney, C. A. Bolle, P. R. Kolodner, R. Ryf, D. T. Neilson, and J. V. Gates, "1100 x 1100 port mems-based optical crossconnect with 4-db maximum loss," *IEEE Photonics Technology Letters*, vol. 15, pp. 1537–1539, Nov 2003.

[17] V. A. Aksyuk, S. Arney, N. R. Basavanhally, D. J. Bishop, C. A. Bolle, C. C. Chang, R. Frahm, A. Gasparyan, J. V. Gates, R. George, C. R. Giles, J. Kim, P. R. Kolodner, T. M. Lee, D. T. Neilson, C. Nijander, C. J. Nuzman, M. Paczkowski, A. R. Papazian, R. Ryf, H. Shea, and M. E. Simon, "238x238 surface micromachined optical crossconnect with 2db maximum loss," in *Optical Fiber Communication Conference and Exhibit*, pp. FB9–1–FB9–3, Mar 2002.

[18] M. Kawai, K. Mori, T. Yamamoto, O. Tsuboi, K. Tanaka, T. Yamamoto, K. Morito, I. Sawaki, and M. Sugawara, "Optical components and devices for next-generation networks," *Fujitsu Sci. Tech. Journal*, vol. 42, pp. 483–493, Oct 2006.

[19] "Lucent announces first all-optical MEMS-based transport device." http://www.lightwaveonline.com/articles/print/volume-17/issue-1/news/lucent-announces-first-all-optical-mems-based-transport-device-53461222.html, January 2000. [Lucent Technologies; Online; accessed 19-August-2017].

[20] "Lucent Launches All-optical Cross-Connect." https://www.fiberopticsonline.com/doc/lucent-launches-all-optical-cross-connect-0001, 2000. [Lucent Technologies; Online; accessed 19-August-2017].

[21] D. J. Bishop, C. R. Giles, and G. P. Austin, "The lucent lambdarouter: Mems technology of the future here today," *IEEE Communications Magazine*, vol. 40, pp. 75–79, Mar 2002.

[22] V. A. Aksyuk, F. Pardo, C. A. Bolle, S. Arney, C. R. Giles, and D. J. Bishop, "Lucent microstar micromirror array technology for large optical crossconnects," *Proc.SPIE*, vol. 4178, pp. 320–324, 2000.

[23] R. Ryf, J. Kim, J. P. Hickey, A. Gnauck, D. Carr, F. Pardo, C. Bolle, R. Frahm, N. Basavanhally, C. Yoh, D. Ramsey, R. Boie, R. George, J. Kraus, C. Lichtenwalner, R. Papazian, J. Gates, H. R. Shea, A. Gasparyan, V. Muratov, J. E. Griffith, J. A. Prybyla, S. Goyal, C. D. White, M. T. Lin, R. Ruel, C. Nijander, S. Arney, D. T. Neilson, D. J. Bishop, P. Kolodner, S. Pau, C. Nuzman, A. Weis, B. Kumar, D. Lieuwen, V. Aksyuk, D. S. Greywall, T. C. Lee, H. T. Soh, W. M. Mansfield, S. Jin, W. Y. Lai, H. A. Huggins, D. L. Barr, R. A. Cirelli, G. R. Bogart, K. Teffeau, R. Vella, H. Mavoori, A. Ramirez, N. A. Ciampa, F. P. Klemens, M. D. Morris, T. Boone, J. Q. Liu, J. M. Rosamilia, and C. R. Giles, "1296-port mems transparent optical crossconnect with 2.07 petabit/s switch capacity," in *OFC 2001. Optical Fiber Communication Conference and Exhibit. Technical Digest Postconference Edition (IEEE Cat. 01CH37171)*, vol. 4, pp. PD28–PD28, March 2001.

[24] "S320 Optical Circuit Switch." http://www.calient.net/members-area/?redirect-to=/download/s320-optical-circuit-switch-datasheet/, 2017. [Datasheet; Calient; http://www.calient.net/products/s-series-photonic-switch/, Online; accessed 19-August-2017].

[25] "Glimmerglass Intelligent Optical System 600." http://www.glimmerglass.com/wp-content/uploads/DS-System-600.pdf, 2017. [Datasheet; Glimmerglass; http://www.glimmerglass.com/index.php/products/intelligent-optical-systems/, Online; accessed 19-August-2017].

[26] "32X32 MEMS 3D matrix optical switch." https://www.diconfiberoptics.com/products/scd0318/SCD-0318B.pdf, 2017. [Datasheet; DiCon Fiberoptics; https://www.diconfiberoptics.com/products/mems_matrix_optical_switches.php, Online; accessed 19-August-2017].

[27] I. Plander and M. Stepanovsky, "Gas-damped double-gimbaled electrostatic torsional micromirror modeling and simulation," *Simulation Modelling Practice and Theory*, vol. 20, no. 1, pp. 59 – 69, 2012.

[28] I. Plander and M. Stepanovsky, "MEMS optical switch: Switching time reduction," *Open Computer Science*, vol. 6, pp. 116 – 125, 2016.

[29] I. Plander and M. Stepanovsky, "Advanced three-dimensional MEMS photonic cross-connect switch for nonblocking all-optical networks," *Optical Switching and Networking*, vol. 22, no. Supplement C, pp. 42 – 53, 2016.

[30] I. Plander and M. Stepanovsky, "Decoupling of two-axis electrostatically-actuated 3D MEMS mirror," in *2015 IEEE 13th International Scientific Conference on Informatics*, pp. 211–216, Nov 2015.

[31] I. Plander and M. Stepanovsky, "The multiphysics model of the gas-damped micromirror for the MEMS-based optical switch," in *MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*, pp. 2700–2703, February 2009.

[32] I. Plander and M. Stepanovsky, "The distributed parameter model of the electrostatic-actuated gas-damped MEMS-based devices for the simulation on parallel computer systems," in *5th International Workshop on Grid Computing for Complex Problems, GCCP 2009*, October 2009.

[33] J. che Tsai, S. Huang, D. Hah, H. Toshiyoshi, and M. C. Wu, "Open-loop operation of mems-based 1 n wavelengthselective switch with long-term stability and repeatability," *IEEE Photon. Technol. Lett*, pp. 1041–1043, 2004.

[34] J. Tsai and M. C. Wu, "A high port-count wavelength-selective switch using a large scan-angle, high fill-factor, two-axis mems scanner array," *IEEE Photonics Technology Letters*, vol. 18, pp. 1439–1441, July 2006.

[35] M. C. Wu, O. Solgaard, and J. E. Ford, "Optical mems for lightwave communication," *Journal of Lightwave Technology*, vol. 24, pp. 4433–4454, Dec 2006.

[36] M. C. M. Lee and M. C. Wu, "Mems-actuated microdisk resonators with variable power coupling ratios," *IEEE Photonics Technology Letters*, vol. 17, pp. 1034–1036, May 2005.

[37] J. Yao, D. Leuenberger, M. C. M. Lee, and M. C. Wu, "Silicon microtoroidal resonators with integrated mems tunable coupler," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 13, pp. 202–208, March 2007.

[38] G. N. Nielson, D. Seneviratne, F. Lopez-Royo, P. T. Rakich, Y. Avrahami, M. R. Watts, H. A. Haus, H. L. Tuller, and G. Barbastathis, "Integrated wavelength-selective optical mems switching using ring resonator filters," *IEEE Photonics Technology Letters*, vol. 17, pp. 1190–1192, June 2005.

[39] G. N. Nielson, D. Seneviratne, F. Lopez-Royo, P. T. Rakich, F. Giacometti, H. L. Tuller, and G. Barbastathis, "Mems based wavelength selective optical switching for integrated photonic circuits," in *Conference on Lasers and Electro-Optics/International Quantum Electronics Conference and Photonic Applications Systems Technologies*, p. CTuFF5, Optical Society of America, 2004.

[40] S. Han, T. J. Seok, N. Quack, B.-W. Yoo, and M. C. Wu, "Monolithic 50x50 mems silicon photonic switches with microsecond response time," in *Optical Fiber Communication Conference*, p. M2K.2, Optical Society of America, 2014.

[41] S. Han, *Highly Scalable Silicon Photonic Switches Based on Waveguide Crossbar with Movable Waveguide Couplers*. PhD thesis, University of California, Berkeley, 2016.

[42] T. J. Seok, N. Quack, S. Han, R. S. Muller, and M. C. Wu, "Large-scale broadband digital silicon photonic switches with vertical adiabatic couplers," *Optica*, vol. 3, pp. 64–70, Jan 2016.

[43] N. Quack, T. J. Seok, S. Han, R. S. Muller, and M. C. Wu, "Scalable row/column addressing of silicon photonic mems switches," *IEEE Photonics Technology Letters*, vol. 28, pp. 561–564, March 2016.

[44] E. Paspalakis, "Adiabatic three-waveguide directional coupler," *Optics Communications*, vol. 258, no. 1, pp. 30 – 34, 2006.

Non-computable Models of Certain First Order Theories

Gábor Sági
Alfréd Rényi Institute of Mathematics,
Hungarian Academy of Sciences,
Reáltanoda u. 13-15,
H-1053 Budapest, Hungary
and
Budapest University of Technology and Economics,
Department of Algebra,
Egry J. u. 1,
H-1111 Budapest, Hungary
Email: sagi@renyi.hu

Ramón Horváth
InstaBridge AB,
111 45 Stockholm, Sweden,
Birger Jarlsgatan 43,
Email: haromn@gmail.com

***Abstract*— Let $\mathcal{D}$ be a complexity class. A countable first order structure is defined to be $\mathcal{D}$-presented iff all of its basic relations and functions are in $\mathcal{D}$. We show, that if T is a first order theory with at least one uncountable Stone space then T has a countable model not isomorphic to any $\mathcal{D}$-presented one. We also show that there is a countable $\aleph_0$-categorical structure in a finite language which is not isomorphic to any $\mathcal{D}$-presented structure; in addition, there exists a consistent first order theory in a finite language that does not have $\mathcal{D}$-presented models, at all.**

Our proofs utilize model theoretic methods and do not involve any nontrivial recursion theoretic notion or construction.

***AMS Subject Classification:* 03C57, 03D45.**
***Keywords:* Computable structures, complexity classes, $\aleph_0$-categorical structures, oligomorphic permutation groups.**

I. Introduction

Abstract data types (for example, in object oriented programming languages) may be regarded as certain first order structures with countable universe and a set of computable operations and relations in it. Further, very often it is important to implement algorithms capable to perform operations in certain countable first order structures (such as algebraic number fields, rings, countably infinite Boolean Algebras, or groups, etc.). Implementing such algorithms have theoretical limitations: all the operations and relations should be recursive in the algorithm theoretical sense. In this paper we investigate such theoretical limitations, the main results are Theorems 4.1 and 5.13; they will be recalled below in the Intorduction after some technical preparations.

A first order structure $\mathfrak{A}$ is defined to be *computable* iff its universe is the set of natural numbers and all of its basic relations and functions are recursive (in the computational theoretic sense). More generally, if $\mathcal{D}$ is a complexity class (like the set of recursive, or the set of recursively enumerable relations), then a countable structure $\mathfrak{A}$ is defined to be *$\mathcal{D}$-presented* iff all of its basic relations and functions belong to $\mathcal{D}$. For a more precise definition we refer to Section II below.

By some classical results of Ershov, Arslanov and others, there are countable orderings, Boolean algebras, etc. which are not isomorphic to any computable structure.

One of the main aims of model theory is to describe all structures in which a given theory (i.e. set of first order formulas) is true. At that level of generality this ambitious aim seems to be untractable. Hence, instead of it, model theorists are trying to characterize those theories which have a structure theorem, that is, whose models can be described in a comprehensive way. Recently, related investigations are very active. Along the results of Morley, Shelah, Lascar, Hrushovski, Cherlin, Pillay and others, it turned out, that theories have a "structure theoretic" hierarchy of complexity: in some cases the possible models are relatively easy to describe, in some other cases this is much more difficult, while in some other cases such a complete "comprehensive" description of

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

all models is impossible for theoretical reasons. This hierarchy is not as exactly defined as the computational theoretic one. In order to make it more precise, one can "measure the structure theoretic complexity" of a first order theory T by the number of pairwise non-isomorphic models of T of a given cardinality, or by different degrees of stability, i.e. by the size of the Stone spaces of the theory (for the definition of Stone spaces we refer to Section II below). It turned out, that the elements of the above list are closely related to each other. Somewhat roughly, but more concretely, $\aleph_0$-categorical theories (which are the simplest ones from structure theoretic point of view) have small (finite) Stone spaces, while the Stone spaces of unstable theories are of large (uncountable) cardinality; if a theory has an uncountable Stone space over the empty set, then it has uncountably many pairwise non-isomorphic countable models.

Some resluts relating model theoretic and recursion theoretic complexity have already been established. For example, the authors in [9] and in [8] deal with countable, computable models of uncountably categorical theories. For further related results we refer the reader to [10] and to [6]; they also contain a rather complete list of references. For more recent related investigations we refer to [5] and [1].

Some of the known results are "positive", that is, they guarantee the existence of a computable model. Negative results state, that certain models do not have recursively presented isomorphic copies (but do not exclude the existence of a $\mathcal{D}$-presented isomorphic copy, where $\mathcal{D}$ is a higher complexity class – like the class of arithmetical relations, for example).

In this paper we are also trying to compare structure theoretic and computational theoretic complexities of first order theories, and we are interested in "strong negative results", that is, our aim is to prove that certain theories have countable models having highly non-computable isomorphic copies, only. In more detail, the main results of the paper are Theorems 4.1 and 5.13; to claim them, let $\mathcal{D}$ be an arbitrary complexity class.

> In Theorem 4.1 we show that if a theory T is complicated in the model theoretic sense (i.e. at least one of it's Stone spaces is uncountable) then T has a "complicated" countable model. More precisely, T has a countable model which is not isomorphic to any $\mathcal{D}$-presented structure.
>
> In Theorem 5.13 we show, that there exist a finite first order language and a theory T in it, which is as simple as possible from the model theoretic sense (namely, T is consistent and $\aleph_0$-categorical, hence all of its Stone-spaces are finite), but the unique countable model of T does not have a $\mathcal{D}$-presented isomorphic copy.

To prove theorem 5.13, it would be enough for us to construct uncountably many pairwise "essentially different" oligomorphic permutation groups on ω. For completeness, we show in theorem 5.11 that there are $2^{\aleph_0}$ many such groups.

There is another related interesting question which we are able to answer. Let $\mathcal{D}$ be a fixed complexity class. As we mentioned, a structure is used to be considered "complicated" from the recursion theoretic point of view, if it is not isomorphic to any $\mathcal{D}$-presented one. It seems to be also natural to replace "isomorphic" with "elementarily equivalent" in the above sentence and to ask whether a first order structure is elementarily equivalent with a $\mathcal{D}$-presented one. For example, if $\mathcal{D}$ is the class of recursive relations and a structure $\mathfrak{A}$ is elementarily equivalent with a $\mathcal{D}$-presented one, then (some fragments of) the theory of $\mathfrak{A}$ may be algorithmically decided. According to our knowledge, there are no previous investigations for structures elementarily equivalent to $\mathcal{D}$-presented ones. In Theorem 5.13 we also show that

> there is a finite language L and an L-structure $\mathfrak{A}$ which is not elementarily equivalent to any $\mathcal{D}$-presented structure (that is, the first order theory of $\mathfrak{A}$ does not have $\mathcal{D}$-presented models, at all).

Our proofs utilize model theoretic methods and do not involve any nontrivial recursion theoretic notion or construction. Since we believe, that the presented results may be interesting both for model theorists and for recursion theorists, we will present more details than as usual. Particularly, for the readers convenience, sometimes we will include known proofs in the present paper.

The structure of the paper is as follows. At the end of the present section we are summing up our system of notation. In Section II we recall the recursion theoretic and model theoretic notions we will dealing with. In Section III we present some observations which we will use in later sections. In Section IV we prove Theorem 4.1: if a theory has an uncountable Stone space, then it has a countable model which does not have a $\mathcal{D}$-presented isomorphic copy. In Section V we are dealing with $\aleph_0$-categorical structures. In Theorem 5.13 we show that there is a finite language L and an $\aleph_0$-categorical L-structure which does not have a $\mathcal{D}$-presented isomorphic copy; in addition, there is a consistent first order theory in L which does not have a $\mathcal{D}$-presented model. In Theorem 5.11 we also show that there are $2^{\aleph_0}$ many closed oligomorphic permutation groups on ω having pairwise different orbit sequences. Finally, in Section VI we present a question that remained open.

Notation

Our notation is mostly standard, but the following list may help.

Throughout ω denotes the set of natural numbers and for every $n \in \omega$ we have $n = \{0, 1, ..., n-1\}$. Let A and B be sets. Then ${}^{A}B$ denotes the set of functions from A to B, $|A|$ denotes the cardinality of A, $[A]^{<\omega}$ denotes the set of finite subsets of A and if κ is an ordinal then ${}^{<\kappa}A$ denotes the set of A-termed sequences of length smaller than κ. If s and t are sequences, then $s^\frown t$ denotes their concatenation.

Throughout we use function composition in such a way that the rightmost factor acts first. That is, for functions f, g we define $f \circ g(x) = f(g(x))$. Structures are denoted by gothic letters, like $\mathfrak{A}$ or $\mathfrak{B}$; the universe of a given structure will be denoted by the same latin letter A or B, respectively. The automorphism group of the structure $\mathfrak{A}$ will be denoted by $\mathrm{Aut}(\mathfrak{A})$.

II. Complexity

We start by recalling the notion of *complexity classes*.

Definition 2.1: Consider two relations on ω, say $R_1 \subset \omega^n$, and $R_2 \subset \omega^m$. We say that R_1 has a *reduction* to R_2, in symbols $R_1 \prec R_2$, iff there exsists a recursive algorithm or map, say M, such that $w \in R_1 \Leftrightarrow M(w) \in R_2$. We say that R_1 is *recursive relative* to R_2.

This definition means that R_2 is at least as a hard problem as R_1. If we have an algorithm solving the membership problem for R_2 then, up to recursivity, it may also be utilized to solve the membership problem for R_1 as well. It is easy to see that $\prec$ is a reflexive and transitive relation. Consequently – as it is well known – the relation $\prec$ determines an equivalence relation $\sim$ via the stipulation:

$$R_1 \sim R_2 \text{ iff } R_1 \prec R_2 \text{ and } R_2 \prec R_1.$$

Definition 2.2: By a *complexity class* we mean an equivalence class of $\sim$ containing relations of arbitrary arity on ω.

These kind of complexity classes are called *Turing degrees* or degrees of unsolvability (see [2]). We denote them by calligraphic letters like $\mathcal{D}$. Typical examples for complexity classes are

- the set of recursive relations;
- the set of recursively enumerable relations;
- the set of arithmetical relations.

(If we had used a reduction with more restrictions we would have obtained a more detailed classification such as Karp–classes, for example.)

We recall the following well known facts from [2].

Lemma 2.3: (1) A complexity class is always countable; (2) The set of all complexity classes has cardinality $2^{\aleph_0}$; (3) Comlexity classes are partially ordered by the induced ordering of $\prec$; (4) A set of complexity classes has an upper bound (under $\prec$) iff it is countable.

Now we recall some model theoretic notions.

Let κ be a cardinal and T a set of first order formulas. Then T is defined to be κ-categorical iff up to isomorphism, T has a unique κ-sized model. A structure is κ-categorical iff its theory is κ-categorical.

Let $\mathfrak{A}$ be a structure, $X \subseteq A$ and $n \in \omega$. Then the n^{th} Stone space $S_n^{\mathfrak{A}}(X)$ of $\mathfrak{A}$ over X is the topological dual space of the Boolean algebra of n-ary relations definable in $\mathfrak{A}$ with parameters from X. Elements of $S_n^{\mathfrak{A}}(X)$ are called types (more precisely, they are called n-types of $\mathfrak{A}$ over X). Similarly, if T is a theory (i.e., T is a set of first order formulas) then the n^{th} Stone space $S_n(T)$ of T is the topological dual space of the Boolean algebra of n-ary relations definable in T; elements of $S_n(T)$ are also called types of T. For further details we refer to [7].

III. Preliminary Observations

Let L be a finite first order language and $\mathcal{D}$ be a complexity class. We say, that an L-structure $\mathfrak{A}$ has a $\mathcal{D}$-presentation, iff there is a structure $\mathfrak{B} = \langle \omega, R_i^{\mathcal{B}} \rangle_{i \in L}$ such that $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic and for every $i \in L$, $R_i^{\mathfrak{B}}$ (as a subset of a direct power of ω) is in $\mathcal{D}$. In this case we also say, that $\mathfrak{B}$ is $\mathcal{D}$-presented.

Throughout this section L denotes a finite first order language.

Lemma 3.1: Let $\mathcal{D}$ be a complexity class. Suppose $\mathcal{H}$ is an uncountable family of pairwise non-isomorphic countable L-structures. Then there exists $\mathfrak{A} \in \mathcal{H}$ such that $\mathfrak{A}$ does not have a $\mathcal{D}$-presentation.

Proof: Let $\mathcal{H}_0 = \{\mathfrak{A} \in \mathcal{H} : \mathfrak{A}$ has a $\mathcal{D}$-presentation $\}$ and for every $\mathfrak{A} \in \mathcal{H}_0$ let $\mathcal{D}(\mathfrak{A})$ be a $\mathcal{D}$-presented structure such that $f_{\mathfrak{A}} : A \to \mathcal{D}(A)$ is an isomorphism between $\mathfrak{A}$ and $\mathcal{D}(\mathfrak{A})$. Observe, that for every distinct $\mathfrak{A}, \mathfrak{B}$ we have $\mathcal{D}(\mathfrak{A}) \neq \mathcal{D}(\mathfrak{B})$, otherwise $(f_{\mathfrak{B}})^{-1} \circ f_{\mathfrak{A}}$ would be an isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$. Hence, the function $\alpha : \mathfrak{A} \mapsto \mathcal{D}(\mathfrak{A})$ is injective. In addition, there are only countably many $\mathcal{D}$-presented L-structures, so the range of α is countable. It follows, that $\mathcal{H}_0$ (which is the domain of α) is also countable. Consequently, there exists $\mathfrak{A} \in \mathcal{H} \backslash \mathcal{H}_0$; this $\mathfrak{A}$ does not have a $\mathcal{D}$-presentation. ■

Corollary 3.2: (1) There is an ordering on ω which does not have a computable presentation.
(2) There is a well-ordering on ω which does not have an arithmetical presentation.

Proof: Since (2) implies (1), it is enough to show (2). Let $\mathcal{H}$ be the set of (isomorphism types of) countable well-orderings and let $\mathcal{D}$ be the set of arithmetical relations on ω. Since $|\mathcal{H}| = \aleph_1$, the statement follows from Lemma 3.1. ■

IV. Theories with many types

As we mentioned in the Introduction, a theory is used to consider "complicated" from model theoretic point of view iff at least one of its Stone spaces is of uncountable cardinality. In the present section we show that if a theory T has at least one uncountable Stone space (that is, if there is $n \in \omega$ with $|S_n(T)| \geq \aleph_1$) then T has a countable model which does not have a computable presentation (even does not have a $\mathcal{D}$-presented isomorphic copy, where $\mathcal{D}$ is an arbitrary fixed complexity class).

Theorem 4.1: Let $\mathcal{D}$ be a complexity class and let T be a first order theory in a finite language such that there is an $n \in \omega$ with $|S_n(T)| \geq \aleph_1$. Then T has a countable model which is not isomorphic to any $\mathcal{D}$-presented structure.

Proof: We apply transfinite recursion. Suppose that we have a countable set of countable structures $\{\mathfrak{A}_\alpha : \alpha < \lambda\}$ with $\lambda < \aleph_1$ such that they are pairwise nonisomorphic and are models of T. Each structure can realize only countably many types from $S_n(T)$, since a single n-tuple realizes a unique n-type. Hence, these countably many structres realize countably many types alltogether. Let us choose a type $p \in S_n(T)$ which has not been realized yet. It follows, that there is a structure $\mathfrak{B}$ which realizes p and is a model of T. The downward Löwenheim–Skolem theorem implies, that there exists a countable elementary substructure $\mathfrak{A}_\lambda$ of $\mathfrak{B}$ such that $\mathfrak{A}_\lambda$ still realizes p. Thus $\mathfrak{A}_\lambda$ has the property: $\mathfrak{A}_\lambda \models T$, $\mathfrak{A}_\lambda \ncong \mathfrak{A}_\alpha$ $(\alpha < \lambda)$. In this way we construct $\aleph_1$ many pairwise nonisomorphic models of T. Hence, the statement follows from Lemma 3.1. ■

Remark 4.2: It is well-known, that $|S_n(T)| \geq \aleph_1$ implies $|S_n(T)| = 2^{\aleph_0}$ (see e.g. theorem 6.3.4 of Hodges [7]).

V. $\aleph_0$-categorical Theories

As we already mentioned, from structure theoretic point of view, a theory T is as simple as possible, iff it is $\aleph_0$-categorical, that is iff T has a unique countable model.

In this section we show, that there exists an $\aleph_0$-categorical theory in a finite language whose unique countable model does not have a computable isomorphic copy (that is, altough T is simple from structure theoretic point of view, its unique countable model is still complicated from computatinal theoretic point of view).

In addition, we also show that there is a countable structure which is not elementarily equivalent to any computable (or any $\mathcal{D}$-presented) structures, where $\mathcal{D}$ is a given complexity class.

We recall a well known result of Svenonius, Ryll–Nardzewski and others which establishes a connection between $\aleph_0$-categoricity and the size of Stone spaces.

Theorem 5.1: For a theory T the following two conditions are equivalent:
(1) T is $\aleph_0$-categorical;
(2) for all $n \in \omega$ we have $|S_n(T)| < \aleph_0$.

The proof can be found in practically every monograph on model theory.

To prove our results first we need to recall some further known connections between $\aleph_0$-categorical structures and certain permutation groups on ω.

Definition 5.2: A permutation group $\mathcal{G}$ acting on the set X is defined to be *closed* iff for every permutation $f \in {}^X X$ the following holds:

if for every finite $s \subseteq X$ there is $g_s \in G$ such that $f|_s = g_s|_s$, then $f \in G$.

Equip X with the discrete topology. Then $\mathcal{G}$ is a closed permutation group iff it is a closed subset of ${}^{X}X$ in the corresponding product topology. For more details we refer to [7].

Clearly, the automorphism group of a first order structure is closed.

Definition 5.3: A permutation group $\mathcal{G}$ on X is said to be *oligomorphic* iff for every $n \in \omega$ the group acts on the n-tuples in a way that the number of orbits is finite.

If $\mathcal{G}$ is an oligomorphic permutation group on X and $n \in \omega$ then $o_n^{\mathcal{G}}$ denotes the number of orbits of $\mathcal{G}$ on the set of n-tuples of X.

Lemma 5.4: If $\mathcal{G}$ is a closed oligomorphic permutation group on ω then there exists an $\aleph_0$-categorical structure $\mathfrak{A}$ on ω with $Aut(\mathfrak{A}) = \mathcal{G}$.

Proof: This theorem is well known, a proof can be reconstructed e.g. by combining Theorems 4.1.4 (b) and 7.3.1 of [7]. ■

Lemma 5.5: For any sequence $\langle a_n \in \omega : n \in \omega \rangle$ there is an oligomorphic group $\mathcal{G}$ for which $o_n^{\mathcal{G}} > a_n$ for all $n \in \omega$.

Proof: The proof can be found e.g. in Cameron [3] (see Item 3.24 therein). ■

We say that two oligomorphic permutation groups $\mathcal{F}, \mathcal{G}$ have same orbit sequences iff for every $n \in \omega$ we have $o_n^{\mathcal{F}} = o_n^{\mathcal{G}}$.

Lemma 5.6: There exist $\aleph_1$ many oligomorphic permutation groups on ω with pairwise different orbit sequences.

Proof: We apply transfinite recursion. Suppose we have $\{\mathcal{G}_\alpha : \alpha < \beta\}$ where β is a countable ordinal, and the $\mathcal{G}_\alpha$'s are oligomorphic permutation groups with pairwise different orbit sequences. So, for every $\alpha < \beta$ we have a sequence $s_\alpha = \langle o_n^{\mathcal{G}_\alpha} : n \in \omega \rangle$. Let $\iota : \omega \to \beta$ be a surjection. Consider the sequence $\langle 1 + o_n^{\mathcal{G}_{\iota(n)}} : n \in \omega \rangle$ as an input for Lemma 5.5. This lemma produces a new oligomorphic group $\mathcal{G}_\beta$ with at least $1 + o_n^{\mathcal{G}_{\iota(n)}}$ many orbits on n-tuples. Finally we obtain $\{\mathcal{G}_\alpha : \alpha < \aleph_1\}$ containing $\aleph_1$ many oligomorphic groups with pairwise different orbit sequences. ■

To prove the main result of the section, Lemma 5.6 would be suitable. For completeness, we show that, in fact, there are $2^{\aleph_0}$ many oligomorphic permutation groups on ω with pairwise different orbit sequences. To do so we need further preparation. We start by recalling two well known lemmas.

Lemma 5.7: If $\mathcal{G}$ is an oligomorphic permutation group on an infinite set then there exists an oligomorphic permutation group on ω with the same orbit sequence.

A proof can be found in subsection 2.2 of [3]. The idea is to build a first order structure from which the group and its action is first order definable, and then apply the downward Löwenheim-Skolem theorem.

Lemma 5.8: Let $\mathcal{G}$ be an oligomorphic permutation group on ω and let $\bar{\mathcal{G}}$ be its closure (in the topological sense). Then

(1) $\bar{\mathcal{G}}$ is an oligomorphic permutation group;

(2) The orbit sequences of $\mathcal{G}$ and $\bar{\mathcal{G}}$ are the same.

Proof: (1) is easy; (2) is straightforward. ■

Next, we present two lemmas we need to prove that there are $2^{\aleph_0}$ many closed, oligomorphic permutation groups on ω with pairwise different orbit serquences.

Lemma 5.9: Suppose $F \subseteq {}^{\omega}\omega$ with $|F| = \aleph_1$ and $s \in {}^{<\omega}\omega$ such that for every $f \in F$ we have $s \subseteq f$. Then there exists $z_0, z_1 \in {}^{<\omega}\omega$ such that $z_0 \neq z_1$, $s \subseteq z_0 \cap z_1$ and

$$|\{f \in F : z_0 \subseteq f\}| = |\{f \in F : z_1 \subseteq f\}| = \aleph_1.$$

Proof: For each $t \in {}^{<\omega}\omega$ let $F_t = \{f \in F : t \subseteq f\}$. First observe, that $F_t = \cup_{n \in \omega} F_{t^\frown n}$, hence

$$(\star) \qquad |F_t| = \aleph_1 \text{ implies } (\exists n \in \omega)(|F_{t^\frown n}| = \aleph_1).$$

Next, assume, seeking a contradiction, that the statement of the lemma is not true. Let $s_0 = s$ and suppose $i \in \omega$ and s_i has already been defined such that $s \subseteq s_i$ and $|F_{s_i}| = \aleph_1$. Then by $(\star)$, and by the indirect assumption, there is a unique $n \in \omega$ such that $|F_{s_i^\frown n}| = \aleph_1$. Let $s_{i+1} = s_i^\frown n$. In this way we defined an increasing sequence $\langle s_i : i \in \omega \rangle$ of finite sequences. Let $f = \cup_{i \in \omega} s_i$. Observe, that for any $t \in {}^{<\omega}\omega$, if $s^\frown t \not\subseteq f$ then $|F_{s^\frown t}| < \aleph_1$ (because otherwise $z_0 = s^\frown t$ and $z_1 = f|_{|s^\frown t|}$ would contradict to our indirect assumption). Hence

$$|\bigcup_{t \in {}^{<\omega}\omega,\ s^\frown t \not\subseteq f} F_{s^\frown t}| \leq \aleph_0.$$

It follows that there exists $g \neq h \in F \setminus \cup_{s^\frown t \not\subseteq f} F_{s^\frown t}$. But then, for each $n \in \omega$ we have $g|_n, h|_n \subseteq f$ which implies $g|_n = h_n$ for each n. This is impossible, since g and h are different functions. ■

Lemma 5.10: Suppose $\langle \mathcal{G}_n, n \in \omega \rangle$ is a sequence of oligomorphic permutation groups such that for every $n \in \omega$ and $k \geq n$ we have $o_n^{\mathcal{G}_n} = o_n^{\mathcal{G}_k}$. Then there is an oligomorphic permutation group $\mathcal{G}$ such that for every $n \in \omega$ we have $o_n^{\mathcal{G}} = o_n^{\mathcal{G}_n}$.

Proof: Suppose $\mathcal{G}_n$ acts on X_n, that is, the elements of G_n are permutations of the set X_n. Let $\mathcal{F}$ be a nonprincipal ultrafilter on ω and let X be the ultraproduct $X = \Pi_{n \in \omega} X_n / \mathcal{F}$. Then $\mathcal{G}$ is defined to be the following permutation group on X:

$$\mathcal{G} \;=\; \{\langle f_n : n \in \omega \rangle / \mathcal{F} :\; (\forall n \in \omega)(f_n \in \mathcal{G}_n)\}.$$

Then the elements of $\mathcal{G}$ are permutations of X, $\mathcal{G}$ is closed under composition and under taking inverses, so $\mathcal{G}$ determines a permutation group. We claim, that $\mathcal{G}$ satisfies the conclusion of the lemma. To see this, let $n \in \omega$ be arbitrary. Then, for every $k \geq n$, one can fix $\{s_i^k : i < o_n^{\mathcal{G}_n}\} \subseteq {}^n X_k$ such that, for fixed k, the s_i^k's lying in pairwise different orbits of $\mathcal{G}_k$. For each $i < o_n^{\mathcal{G}_n}$ let $s_i = \langle s_i^k : k \in \omega, k \geq n \rangle / \mathcal{F}$. It is easy to see, that if $s \in {}^n X$ then there is an $i < o_n^{\mathcal{G}_n}$ such that s and s_i are in the same orbit of $\mathcal{G}$. Hence

$$(1) \qquad o_n^{\mathcal{G}} \leq o_n^{\mathcal{G}_n}.$$

Conversely, if $i \neq j$ then s_i and s_j lie in different orbits of $\mathcal{G}$, which implies

$$(2) \qquad o_n^{\mathcal{G}} \geq o_n^{\mathcal{G}_n}.$$

Combining (1) and (2), the statement follows, as desired. ■

Theorem 5.11: There are $2^{\aleph_0}$ many oligomorphic, closed permutation groups on ω having pairwise different orbit sequences.

Proof: First note, that $|{}^\omega \omega| = 2^{\aleph_0}$, hence there are at most $2^{\aleph_0}$ many permutation groups on ω having pairwise different orbit sequences.

To obtain lower estimation, we show that there is a set $\mathcal{H}'$ consisting of $2^{\aleph_0}$ many oligomorphic, closed permutation groups with pairwise different orbit sequences. To do so, first we build a tree $T = \langle t_s, s \in {}^{<\omega}2 \rangle$ where for every $s \in {}^{<\omega}2$, the elements $t_s \in {}^{<\omega}\omega$ satisfy the following stipulations:

(a) $|t_s| \geq |s|$ and if $s \subseteq s'$ then $t_s \subseteq t_{s'}$;
(b) $s \neq s'$ implies $t_s \neq t_{s'}$;
(c) there are (at least) $\aleph_1$ many oligomorphic permutation groups with pairwise different orbit sequences, such that each of these orbit sequences contain t_s as an initial segment.

Let $t_{\langle\rangle} = \langle\rangle$. By Lemma 5.6 There exist $\aleph_1$ many oligomorphic permutation group with pairwise different orbit sequences, so (c) holds for $t_{\langle\rangle}$. Next, suppose, that $s \in {}^{<\omega}2$ and t_s has already been defined such that (a)-(c) are satisfied. Then, applying Lemma 5.9 we obtain two different extensions t_{z_0} and t_{z_1} of t_s such that (c) remains true if we replace t_s in it by t_{z_0} or t_{z_1}. Let $t_{s\frown 0} = t_{z_0}$ and let $t_{s\frown 1} = t_{z_1}$. In this way (a) and (b) remain true as well, and T can be completely built up.

Now (c) implies, that for every $s \in {}^{<\omega}2$ there exists an oligomorphic permutation group $\mathcal{G}_s$ whose orbit sequence contains t_s as an initial segment. In addition, for every $f \in {}^\omega 2$ let $\mathcal{G}_f$ be the oligomorphic permutation group produced by Lemma 5.10 from the sequence $\langle \mathcal{G}_{f|_n}, n \in \omega \rangle$. Then, $t_{f|_n}$ is an initial segment of the orbit sequence of $\mathcal{G}_f$, for each $n \in \omega$. Hence, if $f, g \in {}^\omega 2$ are different, then there exists $n \in \omega$ with $f(n) \neq g(n)$, so by (a) and (b) $|t_{f|_{n+1}}|, |t_{g|_{n+1}}| \geq n+1$ and $t_{f|_{n+1}} \neq t_{g|_{n+1}}$. In addition, $t_{f|_{n+1}}$ and $t_{g|_{n+1}}$ are initial segments of the orbit sequences of $\mathcal{G}_f$ and $\mathcal{G}_g$, respectively. It follows, that $\mathcal{G}_f$ and $\mathcal{G}_g$ have different orbit sequences. So the set $\mathcal{H}' := \{G_f, f \in {}^\omega 2\}$ consists of oligomorphic permutation groups with pairwise different orbit sequences, as desired.

By Lemma 5.7, for every $\mathcal{G}_f \in \mathcal{H}'$ there exists an oligomorphic permutation group $\mathcal{F}_f$ on ω with the same orbit sequence, by Lemma 5.8 we may assume $\mathcal{F}_f$ is closed as well. ■

Theorem 5.12: There is a finite first order language L in which there are $2^{\aleph_0}$ many pairwise non-isomorphic $\aleph_0$-categorical structures on ω.

Proof: First we recall some facts from [7]. Suppose L is a language containing a distinguished unary relation symbol P and let $\mathfrak{B}$ be an L-structure. We say, that a structure $\mathfrak{A}$ is an induced substructure of $\mathfrak{B}$ by P iff the universe of $\mathfrak{A}$ is $P^{\mathfrak{B}}$ and the definable relations of $\mathfrak{A}$ coincide with the definable relations of $\mathfrak{B}$ restricted to P. This determines $\mathfrak{A}$ up to definitional equivalence, only.

By theorem 7.4.8 of Hodges [7], there is a finite language L containing a distinguished unary relation symbol P such that every $\aleph_0$-categorical structure $\mathfrak{A}$ (possibly having an infinite language) is an induced substructure of an $\aleph_0$-categorical structure $\mathfrak{A}_L$ by P, where the language of $\mathfrak{A}_L$ is L.

By Theorem 5.11 there exists a set $\mathcal{H}' = \{\mathcal{F}_f : f \in {}^\omega 2\}$ of cardinality $2^{\aleph_0}$ containing closed, oligomorphic permutation groups on ω with pairwise different orbit sequences. Lemma 5.4 implies, that for each $\mathcal{F}_f \in \mathcal{H}'$, there is a countable structure $\mathfrak{A}_f$ such that $\mathrm{Aut}(\mathfrak{A}_f) = \mathcal{F}_f$. Then $\mathcal{H}'' = \{\mathfrak{A}_f : f \in {}^{\omega}2\}$ is a set of pairwise non-isomorphic, countable, $\aleph_0$-categorical structures because their automorphism groups are oligomorphic and have pairwise different orbit sequences.

Let $\mathfrak{A}, \mathfrak{B} \in \mathcal{H}''$ be arbitrary, but different. Then they have different orbit sequences, hence $\mathfrak{A}_L$ cannot be isomorphic to $\mathfrak{B}_L$. In other words, the function $\mathfrak{A} \mapsto \mathfrak{A}_L$ is injective on $\mathcal{H}''$. Let $\mathcal{H} = \{\mathfrak{A}_L : \mathfrak{A} \in \mathcal{H}''\}$; clearly $\mathcal{H}$ contains $2^{\aleph_0}$ many pairwise non-isomorphic $\aleph_0$-categorical L-structures, as desired. ■

Theorem 5.13: Let $\mathcal{D}$ be a complexity class.

(1) There exists an $\aleph_0$-categorical structure in a finite language which is not isomorphic to a $\mathcal{D}$-presented structure.

(2) There is a consistent first order theory in a finite language which does not have a $\mathcal{D}$-presented model.

Proof: By Theorem 5.12 there exist a finite language L and a set $\mathcal{H}$ of pairwise non isomorphic, countable $\aleph_0$-categorical L-structures such that $|\mathcal{H}| = 2^{\aleph_0} \geq \aleph_1$. Now (1) follows from Lemma 3.1.

To show (2), let $\mathfrak{A}$ be a structure satisfying (1) and let $T = Th(\mathfrak{A})$. Since $\mathfrak{A}$ is $\aleph_0$-categorical, every countable model of T is isomorphic to $\mathfrak{A}$, hence such a model cannot be $\mathcal{D}$-presented. ■

Remark 5.14: As we mentioned after Lemma 5.6, Theorem 5.13 may be proved more quickly: by Lemma 5.6 there are (at least) $\aleph_1$ many oligomorphic permutation groups on ω having pairwise different orbit structures. By Lemma 5.8 there are (at least) $\aleph_1$ many closed such permutation groups, combining this with Lemma 5.4 and with the technique applied in the proof of Theorem 5.12, we obtain a finite first order language L and $\aleph_1$ many pairwise non-isomorphic, countable $\aleph_0$-categorical L-structures; applying Lemma 3.1 to this family of structures, we also obtain a proof for Theorem 5.13.

VI. Concluding Remarks

The main results of the paper are Theorems 4.1 and 5.13; they can be summarized as follows. Let T be a first order theory and let $\mathcal{D}$ be a complexity class. According to Theorem 4.1 if T has at least one uncountable Stone space then T has a countable model which is not isomorphic to any $\mathcal{D}$-presented structure. Moreover, according to Theorem 5.13, there exists a theory T such that all Stone spaces of T are finite, but still, T does not have a $\mathcal{D}$-presented model.

We conclude this work by noting, that the case of theories with countably infinite Stone spaces is still open. From structure theoretic point of view this case has "intermediate complexity". In general, Lemma 3.1 seems unapplicable for them. However, some striking related results can be found in [9].

Acknowledgment

This work has been supported by Hungarian National Foundation for Scientific Research grant K113047.

References

[1] B. Anderson and B. Csima, *Degrees That Are Not Degrees of Categoricity,* Notre Dame J. Formal Logic Vol. 57 (2016), no. 3, 389–398.

[2] Ed.: J. Barwise, *Handbook of Mathematical Logic,* Studies in Logic and the Found. of Math. Vol 90., Elsevier (2006).

[3] P. J. Cameron, *Oligomorphic Permutation Groups,* LMS 152., Cambridge University Press, Cambridge (1990).

[4] C. C. Chang, H. J. Keisler, *Model Theory,* North–Holland, Amsterdam (1973).

[5] E. B. Fokina, V. Harizanov, A. Melnikov, *Computable Model Theory* pp. 124–194 in: Turing's Legacy: Developments from Turing's Ideas in Logic, Ed.: R. Downey, Cambridge University Press (2014).

[6] V. S. Harizanov, *Computability-theoretic complexity of countable structures,* Bull. Symbolic Logic 8 (2002), no. 4, 457–477.

[7] W. Hodges, *Model theory,* Cambridge University Press, (1997).

[8] B. Khoussainov, M. Laskowski, S. Lempp, R. Solomon, *On the computability-theoretic complexity of trivial, strongly minimal models.* Proc. Amer. Math. Soc. 135 (2007), no. 11, 3711–3721.

[9] B. Khoussainov, A. Nies, R. A. Shore, *Computable models of theories with few models,* Notre Dame J. Formal Logic 38 (1997), no. 2, 165–178.

[10] B. Khoussainov, R. A. Shore, *Effective model theory: the number of models and their complexity,* Models and computability (Leeds, 1997), 193–239, London Math. Soc. Lecture Note Ser., 259, Cambridge Univ. Press, Cambridge, 1999.

[11] D. Marker, *Model Theory. An Introduction,* GTM 217., Springer-Verlag, New York (2002).

Crime Intelligence from Social Media:A Case Study

Serkan Savaş
Yenikent Ahmet Çiçek Vocational and Technical
Anatolian High School
Ankara, Turkey
serkan_savas@hotmail.com

Nurettin Topaloğlu
Gazi University, Faculty of Technology
Ankara, Turkey
nurettin@gazi.edu.tr

***Abstract*- Social media usage is improving every day and new social media websites are being published one by one each day. The huge amount of data generated on social media websites is like a treasure for research and analysis. These data must be processed to reach the information which is aimed. An example application is developed in this study to intelligence on Twitter. Nearly 150 thousand tweets data on Turkish Language are taken from Twitter between specified dates and processed by Turkish Zemberek-NLP (Natural Language Processing) and the relation between data is announced. The crimes are classified according to TUIK (Turkish Statistical Institute) criminals' data and keywords are defined based on these data. Analyze based on these data was applied and results were announced. It is seen that bomb attacks and terror events are spoken on Twitter in Turkey mostly during study dates. Bigger masses can be reachable by expanding keywords group and pulse of community can be listened so, various measures can be taken by doing necessary follows.**

Keywords*- *Big Data; Data Analysis; Intelligence; Social Media; Twitter; Zemberek-NLP

I. INTRODUCTION

Social media usage is improving every day and new social media websites are being published one by one each day. Social media usage is settled among the main objectives of Internet usage today. Facebook, Twitter, YouTube, Instagram, Linked-In, Google+ etc. a lot of social media websites are being used by millions of user actively every day. According to a research, 78% of users group who attended the survey is using internet for social and entertainment aims [1]. The huge amount of data generated on social media websites is like a treasure for research and analysis. According to a report published by Computer Sciences Corporation, the data size will improve 4300% by 2020 compared to today [2]. Internet users are uploading text, image, video and sound data to internet so they are offering many data which includes valuable information by themselves. To reach the information which is aimed, this data must be processed. The huge amount of data with volume, velocity, variety and reality[3] were generated every day. So, the discipline of data mining has begun to shift to the discipline of Big Data. Big data discipline is collaborating with many different disciplines for different purposes. Data is being processed for different purposes and commercial, academic and security issues are main objectives.

Commercial intelligence can be seen anytime by any single user during daily internet usage. After searching a subject in search engines, we are likely to see some offers about that subject in our social media pages. This is related with commercial cyber intelligence as much as big data. For example, "X person searched a book" information is intelligence because the intelligence is considerable information in general. After this intelligence, offering some other books to X person in the Facebook, twitter, etc. means using this cyber intelligence in commerce. Besides this micro commercial cyber intelligence, lots of macro cyber intelligence studies have been made for companies and economy world.

J. Bollen, H. Mao and X. Zeng made a study for estimation of share ratios of Dow Jones Industrial Average (DJIA) companies in 2011 on Twitter. Researchers used daily closing values and people's mood in Twitter so they predicted the daily increase/decrease ratio with %86,7 accuracy [4]. Another similar study was done by X. Zhang, H. Fuehres and P. A. Gloor on Dow Jones, NASDAQ and S&P 500 in 2011. Researchers collected data from Twitter for 6 months and searched the correlation between people's sentiment and shares [5]. R. Agnihotri et al. made a study to show how social media can be used in sales and marketing in 2012 [6].

The academic cyber intelligence work is to open up new studies with the knowledge gained by analyzing the data in the cyber environment and to show the potential of the data flowing in the cyber world.

X. Tang and C. C. Yang offered two-stages system to find secret information inside social media data and tested their technique on three different testbeds with results identifying a high performance level when compared to the baseline algorithm [7]. Another system offer was made by A. Weichselbraun, S. Gindl and A. Scharl in 2014 to discover the meaningful information with correct sentiment analysis inside the social media [8].

M. Atzmueller made a study to find out important structures concern subgroups emerging in those applications as communities, roles and key actors in the networks and communities and opinions, beliefs, and sentiments of the set of actors with data mining in social media [9]. S. Poria et al. introduced a novel paradigm to concept-level sentiment analysis that merges linguistics, common-sense computing and machine learning for improving the accuracy of tasks

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

such as polarity detection in 2014 [10]. S. Bashir, F.H. Khan and U. Qamar presented an algorithm for twitter feeds classification based on a hybrid approach to overcome some limitations on Twitter sentiment analysis [11].

It is obvious that academic social media studies will improve parallel to user numbers and interactions on social media websites. One of the most important studies nowadays is cyber intelligence for security aims. After the reflection of social networks in daily life, individual, institutional and state security issues have emerged at cyber level. In the virtual world that is transforming into the real life itself, there are similar dangers which individuals and institutions encounter in real life. Cyber intelligence is different in terms of state security. This difference can be as preventing the attacks and infiltration or prediction, diagnosis and prevention of possible probabilities in advance by cyber intelligence.

In 2012, Sir D. Omand, J. Barlett and C. Miller announced a new intelligence type called Social Media Intelligence with their study. They explained how to do social intelligence in their study [12]. G. Bayraktar made a study to explain that cyber-attacks can be classified as a new type of war, because it threatens national securities and happens in cyber space that does not have any defined borders in international relations. Also the study defines cyber intelligence concept and its operational domain as well as it puts forth the methodology for consideration in order to gain the initiative in cyber-attacks and to make analysis on the threat [13].

The effects on the mass of social media sites have begun to be scrutinized more carefully after the events of the Arab Spring in the World and researchers made studies to understand the effect of social media and how governments and states took part in this subject [14]. O. Oh, M. Agrawal and H. R. Rao studied citizen-driven information processing through Twitter services using data from three social crises: the Mumbai terrorist attacks in 2008, the Toyota recall in 2010, and the Seattle café shooting incident in 2012 [15]. Another study about Mumbai to build a model of the behavioral pattern by swarm intelligence, a computational intelligence technique was made by B. Chakraborty and S. Banerjee [16].

P. T. and Bertot, J. C. explored the considerations and challenges of the approach, such as the inclusion of members of the public with limited access to the internet, the use of non-governmental channels to disseminate government information, the permanence of digital-born government information, and the design of e-government [17].

An example application is developed in this study to intelligence on Twitter. Nearly 150 thousand tweet data on Turkish Language is taken from Twitter between specified dates and processed by Turkish Zemberek-NLP (Natural Language Processing) [18] and the relation between data is announced.

II. TWITTER INTELLIGENCE

Twitter is one of the leader website which big data applications are used on. It is getting attention of researchers with millions of active users and big amount data produced by these users. Figure 1 shows the graphic of data produced on Twitter.

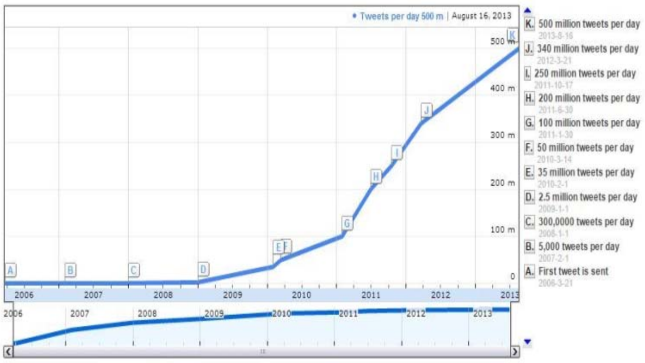


Figure 1. Yearly tweet numbers table [19]

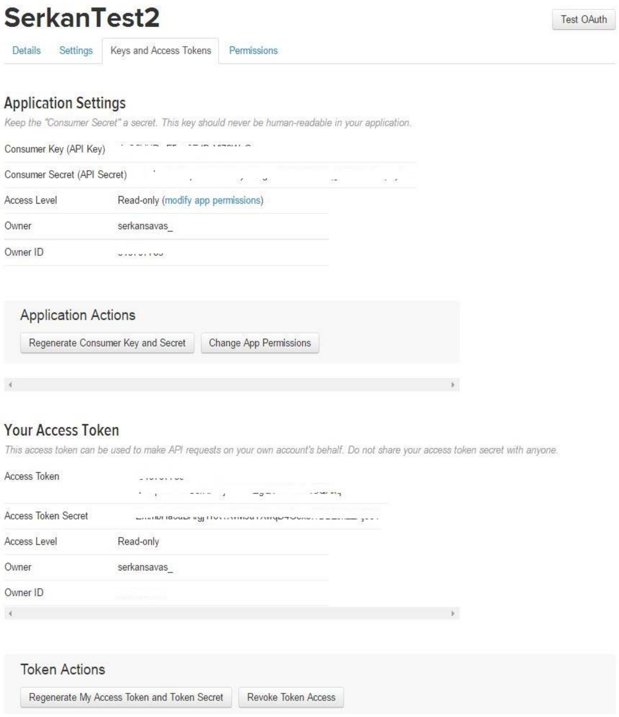
Twitter supports researchers with different APIs (Application Programming Interface). This situation is like win-win for both researchers and Twitter. Researchers have opportunity to investigate different relations between users and Twitter is making advertisement and promotion worldwide with these researchers. Twitter is developed REST API, Stream API and Ads API for researchers to allow them to use these APIs for different purposes.

The REST APIs provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more can be done by REST API. The REST API identifies Twitter applications and users using authentication. The data responses are available in JSON (JavaScript Object Notation) [20-21]. The Streaming APIs give developers low latency access to Twitter's global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint. Twitter offers several streaming endpoints like Public, User and Site and each customized to certain use cases. The Ads API program enables businesses to create and manage ad campaigns programmatically on Twitter. There are types of Ads API such as Standard, Basic and Developer [20].

III. APPLICATION ON TWITTER

There are some steps to use Stream API.

- To create a user account on Twitter.
- To create a new application on Twitter for user. Every user can create applications at https://apps.twitter.com/ address.
- To fill the information for new application and generate keys and auth values of new application (Figure 2).



SerkanTest2

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	[redacted]
Consumer Secret (API Secret)	[redacted]
Access Level	Read-only (modify app permissions)
Owner	serkansavas_
Owner ID	[redacted]

Application Actions

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	[redacted]
Access Token Secret	[redacted]
Access Level	Read-only
Owner	serkansavas_
Owner ID	[redacted]

Token Actions

Figure 2. Applications details, keys and auth information

After creating the own Twitter application, a model is created to work with the data which is taken from Twitter. Flowchart of study is shown in Figure 3.

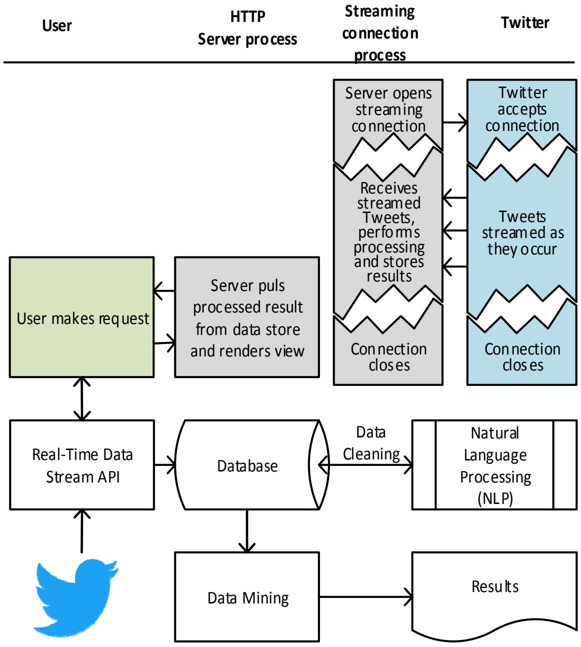

User
HTTP Server process
Streaming connection process
Twitter
Server opens streaming connection
Twitter accepts connection
Receives streamed Tweets, performs processing and stores results
Tweets streamed as they occur
Connection closes
Connection closes
User makes request
Server puls processed result from data store and renders view
Real-Time Data Stream API
Database
Data Cleaning
Natural Language Processing (NLP)
Data Mining
Results

Figure 3. The flowchart of the model in study

In this model, the request was made from Twitter to stream real-time data. During this request, Twitter controls the key and auth values of user API. After control process, Twitter opens connection to the client so client starts to receive and store the tweets. The tweets are stored to a txt file for studying with NLP tools and announcing the results.

A. Data From Twitter

Python programming language and Tweepy libraries have been used to get the data from Twitter. Data transfer on Twitter is listened and the data is get by Tweepy by using filter. Turkish language filter and nine different words are used for real-time data.

For TUIK (Turkish Statistical Institute) data in 2013, criminal cases in Turkey are classified as: assault, theft, opposition to distraint, drug production, weapons crimes, murder, forgery, threatening, spoil, violation of protection of the family, sexual crimes, insult, damage to property, smuggling, using drugs, fraud, restrictions on freedom of the people, taking out of job, traffic offenses, forest crimes, debit, bribe, opposition to military criminal law, mistreatment and other crimes [22].

Bomb, terror, custody, attack, demonstration, smuggling, drugs, prostitution and rape words are chosen as keywords based on these crime classes and recent events in Turkey. Flowchart of the "stream.py" program is shown in Figure 4.

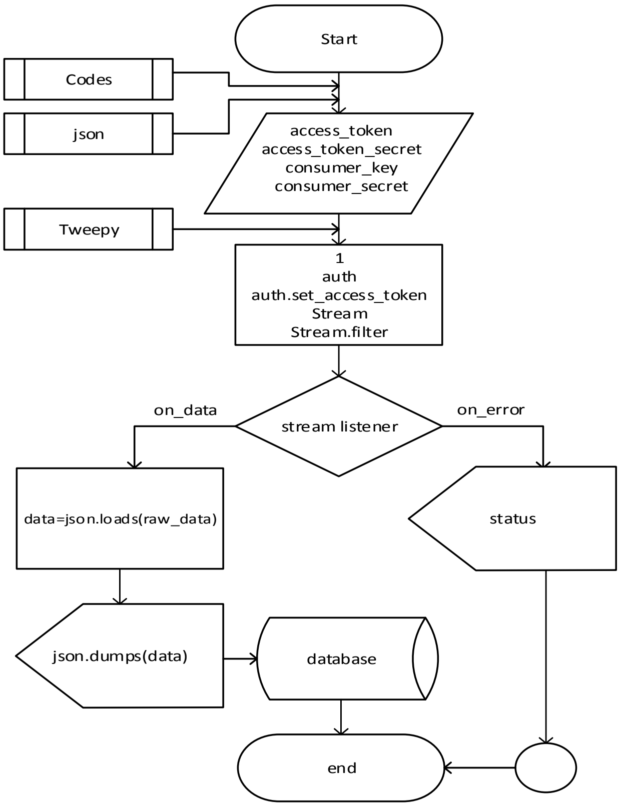


Figure 4. Flowchart of program

Twitter authorization occurs in "stream.py". Also, different libraries as JSON and Tweepy were added to program to filter the tweets and save data as JSON. If Twitter responds to the request the data is saved in a txt file, but if the connection is broken, the error status is given to the client.

IV. RESULTS

Nearly 900 MB size, 158.463 tweet data is taken from Twitter from 3 May 2016 to 5 June 2016. There are 2.101.550 words and 19.074.070 characters in this tweet text group. Zemberek-NLP which is one of the most used NLP libraries for Turkish Language is used to analyze these tweet data.

Zemberek-NLP provides basic statistical tools for letters, words, roots etc. Also its MPL license allows anybody to use anyway. In the beginning, NLP started for Turkish Language but then it widened itself for Turkic Languages [23].

Twitter sends much information with the data taken by Stream API. User name, user ID, time zone, time information, location information, retweet count, favorite count etc. information is among this information. This data can be analyzed and much important information can be reached like a mosaic. These analysis activities can be done with different methods and techniques such as machine learning methods and techniques. NLP tools are also one of the most commonly used tools on Twitter. After data analysis with Zemberek-NLP, words are divided to roots. There are 301.690 roots in data. In this root words, the most frequently seen 20 roots are shown as graphic in Figure 5.

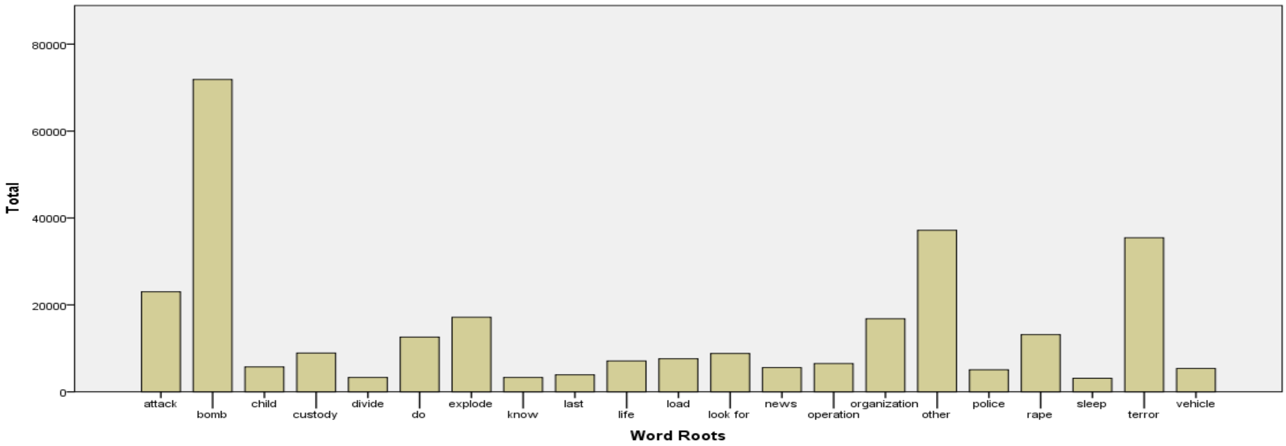


Figure 5. Words roots

In the roots of words, bomb root is the most frequent word and it is followed by terror and attack. Even these three words, can give an idea about analysis. During study days, people have talked about terror bomb attack criminal cases. Other roots include; police station, martyr, eye, woman, look, day, struggle, country, İstanbul, claim, terrorist, year, Suruç, wound and Diyarbakır. These roots have %1 percentages each. After this bar graphic, words without keywords are given in Figure 6.

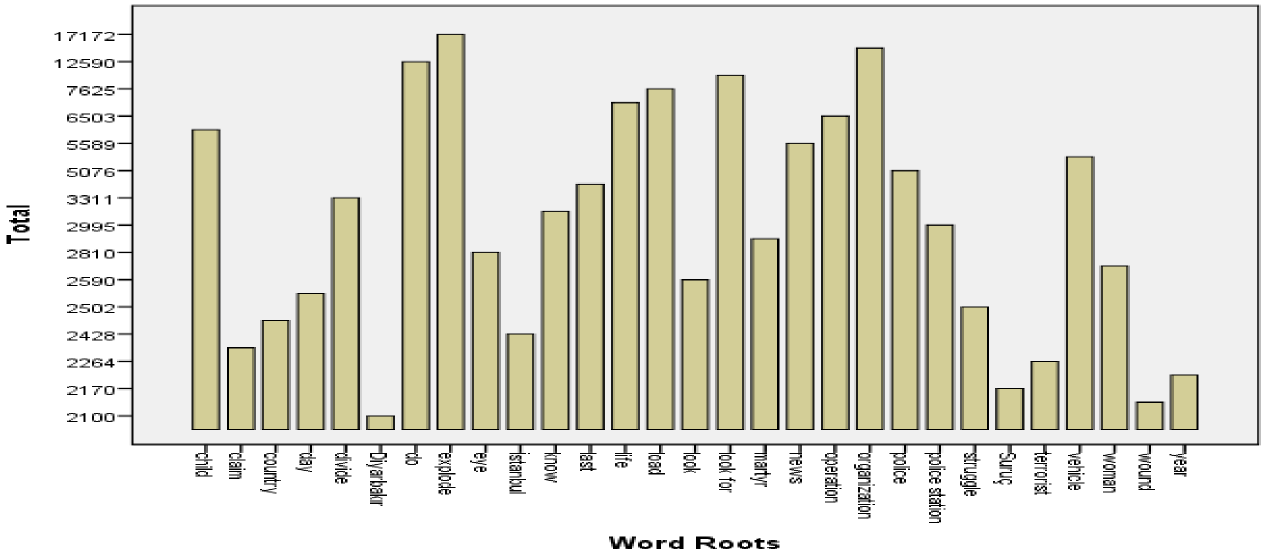


Figure 6. Word roots without keywords

In word roots graphic, the words are translated to English from Turkish. In these keywords, "bomb" was the most frequent word. After that; terror, attack, organization and explode words followed respectively. In this case, it can be understood that during study dates from May to June, mostly terror attacks were spoken in Turkey. It can be seen in figure that which criminal cases are in agenda of the country. Another important issue is no word root from demonstration, smuggling and prostitution keywords. So during real-time data mining, if same situation can be seen in the future works, keywords can change dynamically according to the criminal cases to get more inside the cases. So intelligence will become narrower from wide window. After this graphic, the Zemberek-NLP is applied the data without keywords again. There are 146.149 word roots in the data without keywords. In Figure 6 graphic, explode and organization were the most frequent words. After that; do, look for, life and load words were placed respectively. Similarly with the previous graphic, it can be seen in this graphic that terror attacks are spoken by public. Without any knowledge of tweets data and without seen tweets, from these graphics it can be understood that, terror attacks happened by bombs and vehicles and lives are lost during these attacks.

V. DISCUSSIONS AND SUGGESTIONS

Huge amount of data are released to internet every day with social media usage. This data is being analyzed and people are using this data for different purposes. Commercial, academic and security issues are main purposes for these analyze activities. Twitter is very important social media website in terms of keeping the pulse of the community. It can be listen real-time from Twitter that how people reacts to events, what people talks about, which criminal cases have most attention etc.

This study is an example to analyze studies on Twitter. Based on Turkish Statistical Institute data, criminals are classified and keywords are defined according to these classes. Twitter data have been taken for nearly one month and analyzed. It is seen that bomb attacks and terror events are spoken on Twitter in Turkey mostly during study dates. With widening keyword groups, the more masses can be reachable or by narrowing keywords, some specific issues can be investigated inside. So, researchers can listen to the pulse of community and also government officials can take some precautions by following necessary groups.

REFERENCES

[1] Savaş, S., Topaloğlu, N., & Güler, O. (2015), The Determination of User's Preferences on Some Domain Names in Turkey: A Survey Application, International Journal of Informatics Technologies, vol. 8 (2), 51-58.

[2] Setty, K., & Bakhshi, R. (2013), What is Big Data and What Does It Have to do With It Audit? ISACA Journal, 23-25.

[3] P. C. Zikopoulos, C. Eaton, D. deRoos, T. Deutsch, and G. Lapis, Understanding Big Data, IBM, 2012.

[4] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter Mood Predicts The Stock Market. Journal of Computational Science, 1-8.

[5] Zhang, X., Fuehres, H., & Gloor, P. A., COINs2010: Collaborative Innovation Networks Conference. Predicting Stock Market Indicators Through Twitter "I hope it is not as bad as I fear". Procedia - Social and Behavioral Sciences 7-9 October 2010, Savannah, Georgia, USA

[6] Agnihotri, R., Kothandaraman, P., Kashyap, R., & Singh, R. (2012). Bringing Social into Sales- The Impact of Salespeoples Social Media Use on Service Behaviors and Value Creation. Journal of Personal Selling & Sales Management, 333-345.

[7] Tang, X., & Yang, C. C. (2014). Detecting Social Media Hidden Communities Using Dynamic Stochastic Blockmodel with Temporal Dirichlet Process. ACM Transactions on Intelligent Systems and Technology, 36.

[8] Weichselbraun, A., Gindl, S., & Scharl, A. (2014). Enriching Semantic Knowledge Bases For Opinion Mining In Big Data Applications. KnowledgeBased Systems, 75-85.

[9] Atzmueller, M., (2012), Mining Social Media: Key Players, Sentiments And Communities, Data Mining and Knowledge Discovery, 2: 411–419.

[10] Poria, S., Cambria, E., Winterstein, G., Huang, G-B., (2014), Sentic Patterns: Dependency-Based Rules for Concept-Level Sentiment Analysis Knowledge-Based Systems, 69, 45–63.

[11] Khan, F.H., Bashir, S. & Qamar, U., (2014), TOM: Twitter Opinion Mining Framework Using Hybrid Classification Scheme, Decision Support Systems, 57, 245–257.

[12] Omand, S., Bartlett, J., & Miller, C. (2012). Introducing Social Media Intelligence (SOCMINT). Intelligence and National Security, 801-823.

[13] Bayraktar, G., (2014), The New Requirement for the Fifth Dimension of the War: Cyber Intelligence (Harbin Beşinci Boyutunun Yeni Gereksinimi: Siber İstihbarat), Journal of Security Strategies (Güvenlik Stratejileri Dergisi), 20, 119-147.

[14] Kırık, A. M., (2012), Arab Spring Centext Social Media-Individual Interaction and Social Transformation (Arap Baharı Bağlamı'nda Sosyal Medya-Birey Etkileşimi ve Toplumsal Dönüşüm), Education And Society In The 21st Century The Journal Of Education Science And Social Research (21. Yüzyılda Eğitim Ve Toplum Eğitim Bilimleri Ve Sosyal Araştırmalar Dergisi), 1 (3), 87-98

[15] Oh, O., Agrawal, M. & Rao, H.R., (2013), Community Intelligence and Social Media Services: A Rumor Theoretic Analysis of Tweets During Social Crises, MIS Quarterly, 37 (2), 407-426.

[16] Chakraborty, B. & Banerjee, S., Modeling The Evolution Of Post Disaster Social Awareness From Social Web Sites, 2013 IEEE International Conference on Cybernetics (CYBCO), 13-15 June 2013

[17] Bertot, J.C. & Jaeger, P.T., (2010), Transparency And Technological Change: Ensuring Equal And Sustained Public Access To Government İnformation, Government Information Quarterly, 27, 371–376.

[18] Zemberek-NLP. [Online]. Available: https://zembereknlp.blogspot.com.tr

[19] Internetlivestats. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/#trend

[20] Twitter. [Online]. Available: https://dev.twitter.com/

[21] Robu, D., Sandu, F., Petreus, D., Nedelcu, A., Balica, A., (2014), Social Networking of Instrumentation – a Case Study in Telematics, Advances in Electrical and Computer Engineering14(2), 153-160.

[22] TÜİK- (Turkish Statistical Institute) . [Online]. Available: http://www.tuik.gov.tr/VerBilgi.do?alt_id=1070

[23] C. Çöltekin, A Freely Available Morphological Analyzer for Turkish. LREC. 2010.

Synthesis Criterion of Ergatic Base Complex with Focus on its Reliability

Alena Novák Sedláčková
Department of Air Transport, Faculty of Operation and Economics of Transport and Communications
University of Zilina
Žilina, Slovakia
alena.sedlackova@fpedas.uniza.sk

Pavol Kurdel
Department of Avionics, Faculty of Aeronautics
Technical University in Kosice
Košice, Slovakia
pavol.kurdel@tuke.sk

Boris Mrekaj
Department of Avionics, Faculty of Aeronautics
Technical University in Kosice
Košice, Slovakia
boris.mrekaj @tuke.sk

Abstract— **The paper focuses on the area of economic evaluation and reliability of ergatic base complex operator. The estimation of operation quality, mainly with in-built complex systems, is evaluated with the use of various methods. The selected method presumes the use of efficiency indicator of performed tests in order to prove implementation tests reliability of new complex systems into ergatic base complex. The outputs create numeral characteristics according to which it is possible to estimate the level of conformity of costs on their operation itself. Therefore, the optimization of financial costs on maintaining the reliability of the above mentioned systems with the assumption of estimated level of the optimization of unexpected costs for the operator, e.g. of airport base system, is the decisive parameter [1]. The paper describes the possibilities of MATLAB environment applicability while observing effective use of resources to sustain high reliability of airport base system and its mutual relationship with current complex systems.**

Keywords— ***Matlab, total probability, effectiveness, reliability, ergatic base complex, quadratic criterion, airport base system***

I. Introduction

The calculation of the estimation of ergatic base complexes affectivity at their technical tests is connected with a range of specific particularities, the character of which is determined by their technical specifications and quality. To fulfil the quality of airport base systems, particular methods for their optimal control in the area, by which they can also be evaluated, are selected. The state of environment in which they exist and control abilities of operators with the determined skill is also a condition of effective evaluation of their quality [2]. The main specification at the determination of the beginning of quality is their functionality and the price, which can be evaluated by determined tests of their effective operation. Here minimal number of tests on the base of airport systems in airport base system is desired. Considering the complexity of airport base systems and the need of their *„tuning"(*integrity of all technical systems) is necessary to perform the inspection of relevant parts of airport base system or the effectiveness of construction adjustments [3]. The information, which present test output, possess different statistic character which is analyzed in a complex way. The above mentioned facts limit the use of the calculations of the estimations of experiment effectiveness by classical methods. The complexity of ergatic base systems also requires the use of other calculation methods, with which it is possible to estimate the system effectiveness on the base of a priori information and compare it with the values (estimation) of previous tests [4], [5].

II. Selected Method for Identifying Reliability for Ergatic Base Complexes

In concord with previously made considerations, *time and price* are important parameters, by which ergatic base complexes validity, or ergatic system, is determined. Validity is determined by entered effectiveness and depends on many factors, among which effectiveness models belong as well. Model preciseness is important due to price. How precise the mathematical model is, so small is the number of necessary technical inspections for real tests of technical systems [6]. Simple proportionality which has been presented is important mainly at the models, by which effectiveness dynamics is expressed. This is the reason, why the knowledge of the model construction is an essential factor of analysis of results obtained in the course of all tests which need to be performed to cover qualitative estimation of ergatic base complex functionality. The creation of precise mathematical model of effectiveness dynamics depend on the program and methodology of tests. Majority of models is designed from empirical equations. This fact presents difficulties for the implementation of measured data (e.g. navigation ergatic systems at the airport) into model

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

parameters. In some cases it is almost impossible. This is the reason why the following procedure is usually used [5].

Model structure is determined *apriori (lat.)* and then model parameters are estimated on the base of test results and is performed the control of its adequacy. Preciseness of such a model depends on many items of statistic information, the volume of which can be processed. Generally, volume of measured information obtained during measurements is proportional therefore the focus is on dominant values, from which a fitting set is created. The following step is the selection of effectiveness calculation method, by which the images of the laws of its evolution without requiring high preciseness are created. In this process, the first step is the method, which does not need concrete analytically described model and through which only law trend can be observed [5].

III. EVALUATION FAILURES OF THE ERGATIC BASE COMPLEX

Any phenomena, processes, facts, which exist or are in progress independently of the observer, can be labeled a scientific fact. From logical and epistemological viewpoints, fact is understood as a proved knowledge obtained by the description of selected fragments (ergatic base complex).

Let the program of concrete inherent test effectiveness suppose *N* periods, in each of which a set of identical objects of airport is tested. According to the results of tests, defects which have occurred are recorded and they are removed. Such a procedure secures that each following test has equal input precondition and each is independent from the previous one. In each stage during the test, accidental failures $g_{\ i}$ can occur as the manifestations of structural failures f_i. The number of successful tests is marked m$_i$ and general number of tests n_i.

Let the probability of the manifestation of accidental failure be marked q$_{0,}$ which will be considered a constant valid for all stages. The probability of the manifestation of construction defects in the *i-th* stage is marked q_i. In this case, the probability of failure-free operation, which is named W_i effectiveness, where index *i* presents the number of the stage, will be [4]:

$$W_i \ = 1 - \ q_0 - \ q_i = 1 - (q_0 + \ q_1) \qquad (1)$$

The entry (1) is the analogy of the entry of loss function [1] for *t = 0*. For the effectiveness estimation, the entry has the importance of independence of each following test. Following the above stated facts it is true that as a result of determined number of construction defects in measured ergatic base complexes (if their number does not exceed the determined number), the quantity qi is not growing, i.e.:

$$q_1 \ \geq q_2 \ \geq \cdots \ldots q_i, \ \mathrm{i} = 1{,}2, \ldots\ldots. \mathrm{n} \qquad (2)$$

Fulfilling the condition (2) proves that W_i inherent effectiveness is a not decreasing value.

Effectiveness estimation in the first approximation is performed by the method of maximum effectiveness estimation $W_{max} = m/n$, where *m* is the number of successful tests and *n* – is the number of all performed tests which is shown by the function of maximum probability of the observed case, for the model of functional effectiveness of the whole complex (elements are aligned in series [1]), monitored cases with rate of failures g_i on the *i –th* phase in the following form:

$$L\ (g_1 f_1 m_1, g_2 f_2 m_2\ \ldots\ldots.; \frac{q_n f_n m_n}{q_0 q_1} \ldots . q_n = \prod_{i=}^{n} \frac{n_i!}{\varepsilon_i! f_{\ i}! m_i!}\ q_0^{q_i} q_i^{f_{\ i}} (\,1 - \ q_0\ q_i)^{m_i} \qquad (3)$$

The estimation of maximum value of probabilities q_0 and q_i can be obtained by continuous derivation (3) and comparison with zero.

Step by step the following form is obtained:

$$q_0 \ = \frac{\sum_{i=1}^{n} q_i}{\sum_{i=1}^{n} n_i} \qquad (4)$$

$$q_i \ = \frac{(1 - \ q_0) f_i}{f_i + m_i} \qquad (5)$$

The estimation (4), (5) have been determined regardless of the condition (2). The justness of the condition (1) is evaluated. If this condition is not fulfilled in any stage, i.e. if it is not true that $q_i \ < \ q_{i+1}$, then i-*th* and *(i +1)* stages connect. The quantity qi for such a connection is determined according to (5). The connection of stages goes on until the fulfilment of the required condition: $q_i \geq q_{i+1}$. Then, the estimation of system effectiveness in concord with the dispersion, when it is true that:

$$E\ [D_2^2 - \ D^2] \ < E\ [D_1^2 - \ D^2] \qquad (6)$$

Where: E is medium value of test, D dispersion of test sequence: i = 1, 2, …. .

Effectiveness has been determined according to situational frames, the manifestation success or failure has been evaluated by dichotomy states :(q_i *= 0 - Failure, resp. 1 - Success*).

A. Maximum probability function

The reported process shows the results of the practical measurement in the MATLAB program environment according to (3), (4), (5). Structural failures have been eliminated after modifications to the object structure but also after performing tests in a laboratory environment, see data (g_i, q_0). The recorded number of successful and unsuccessful tests was performed in stages that are expressed on the *x* axis of the graph, see fig. 1. $\alpha^2 + \beta^2 = \chi$. (1) (1)

gi=[0 0 0 1 0 0 0 0 1 4]; vector of accidental quantities evoked by technical environment of base complex (airport).

q0=0.12; influence of external conditions. Constant at all tests of a new system.

mi=[1 0 0 0 3 0 3 2 9 10]; number of successful tests in the period.

fi=[1 1 1 1 1 1 1 1 1 1]; number of removed failures at tests (functionality).

ni=[2 1 1 2 4 1 4 3 11 21]; number of tests.

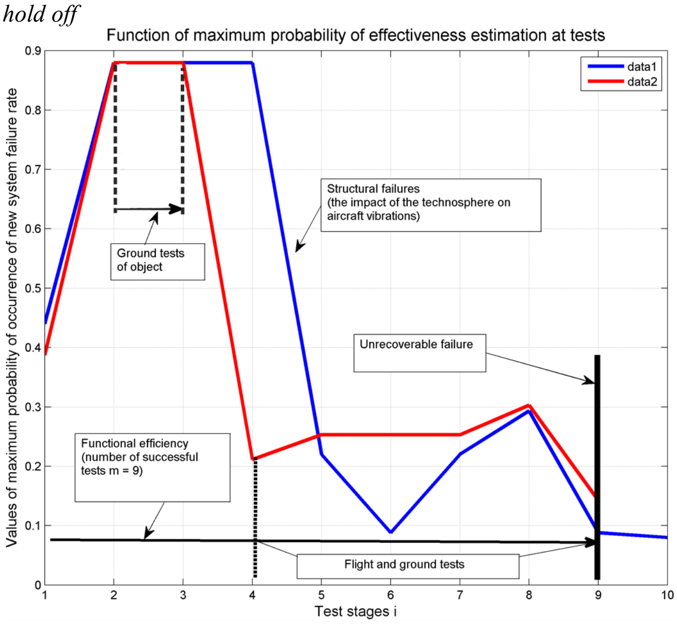
*qi=(1-q0).*fi./(fi+mi);* probability of design defect (1:5:32).

*L=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi. *qi.^fi.*(1-q0-qi).^mi));*

Function of maximum probability. Order of tests:

*i=1;gi=[0];;mi=[1];fi=[1];ni=[2];qi=(1-q0).*fi./(fi+mi),*
*L1=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi .*qi.^fi.*(1-q0-qi).^mi));*
*i=2;gi=[0]; q0=0.12; mi=[0];ni=[1];qi=(1-q0).*fi./(fi+mi),*
*L2=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi .*qi.^fi.*(1-q0-qi).^mi));*
*i=3;gi=[0];q0=0.12;mi=[0];fi=[1];ni=[1];qi=(1-q0).*fi./(fi+mi),*
*L3=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi*

```
.*qi.^fi.*(1-q0-qi).^mi));
i=4;gi=[1];mi=[0];fi=[1];q0=0.12;;ni=[2];qi=(1-q0).*fi./(fi+mi),
L4=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi
.*qi.^fi.*(1-q0-qi).^mi));
i=5;gi=[0];q0=0.12;mi=[3];fi=[1];ni=[4];qi=(1-q0).*fi./(fi+mi),
L5=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi
.*qi.^fi.*(1-q0-qi).^mi));
i=6;gi=[0];q0=0.12;mi=[0];fi=[1];ni=[1];qi=(1-q0).*fi./(fi+mi),
L6=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi
.*qi.^fi.*(1-q0-qi).^mi));
i=7; gi=[0];q0=0.12;mi=[3];ni=[4];fi=[1];qi=(1-q0).*fi./(fi+mi),
L7=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi
.*qi.^fi.*(1-q0-qi).^mi));
i=8;gi=[0];q0=0.12;mi=[2];fi=[1];ni=[3];qi=(1-q0).*fi./(fi+mi),
L8=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi
.*qi.^fi.*(1-q0-qi).^mi));
i=9;gi=[1];q0=0.12;mi=[9];fi=[1];ni=[11];qi=(1-q0).*fi./(fi+mi),
L9=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^gi
.*qi.^fi.*(1-q0-qi).^mi));
i=10;gi=[4];q0=0.12;mi=[10];fi=[1];ni=[21];qi=(1-
q0).*fi./(fi+mi),
L10=(factorial(ni)./(factorial(gi).*factorial(fi).*factorial(mi)).*(q0.^
gi.*qi.^fi.*(1-q0-qi).^mi));
i=1:1:10; Entry of calculated data into table:
q=[0.4400 0.8800 0.8800 0.8800 0.2200 0.0880 0.2200 0.2933
0.0880 0.0800];
L=[0.3872 0.8800 0.8800 0.2112 0.2530 0.2530 0.2530 0.3029
0.1424 NaN];
[i;q;L], Tale of maximum probability.
plot(i,q,i,L),hold on,
xlabel('Test stages i','FontSize',12),
ylabel('Values of maximum probability of occurrence of new system
failure rate','FontSize',12'),
grid on,
title('Function of maximum probability of effectiveness estimation at
tests','FontSize',14'),
hold off
```



Legend: data1 – q_i , (equation 5), data2 – function of maximum probability (equation 3)

Fig. 1. Function of maximum probability of effectiveness estimation at flight tests

Figure 1 shows the fact that at the determination of complex inherent effectiveness of base complex the number of tests is = *9* behaves in a stable way at every following test. This method provides the possibility to designate the behaviour of ergatic base complexes at their implementation into the environment of airport at which its technical effectiveness (technogenic functionality) needs to be determined. When we use the statistic methods, see (window) presented in Matlab, it is possible to evaluate the succession of different processes in the test stage in a quantitative way [7].

IV. Estimation of the Cost of the Tests Performed on the Ergatic Base Complex

By selective method, a dominant, own sub-system is chosen from the object, which considerably influences E-safety (e.g. landing navigation systems). Financial costs to maintain its reliability and energy consumption require constant financial readiness of the operator (e.g. airport). Failure to fulfil this condition creates presumption to create *E-disequilibrium*. Let certain technical equipment marked *C* and its elements $c_{i,j}$ $(i \neq j)$, which create a matrix together. Elements $c_{i,j}$ have importance of capital inputs. To operate *C*, financial means X are necessary, which separately cover the energy consumption and economics of reliable work of elements c_{ij}. The matrix X is a bar one, its elements are $x_{i,j}$, where j=const. Then the volume of financial costs is: $O = C * X$.

Let d present a set of unexpected reasons, which influence O by the element $d_{i,j}$, $j = const.$ as for the capital. The airport operator realizes covering necessary financial costs by matrix A with elements $a_{i,j}$ with required volume: $A * X = b$. Value dimension of the volume b is limited from the bottom *lb* and from the upside *ub*.

The task is to optimize *X* so the condition of quadratic criterion is fulfilled: *||CX-d||^2<_AX=b;*

Left side of the equation presents quadratic criterion of deviation of financial costs control. MATLAB determines: *lsqlin*=||*C*X-d*||.

Following financial volumes present hypothetical estimations.

Matrix *C* of capital inputs:

C = [0.9501 0.7620 0.6153 0.4057; 0.2311 0.4564 0.7919 0.9354; 0.6068 0.0185 0.9218 0.9169; 0.4859 0.8214 0.7382 0.4102; 0.8912 0.4447 0.1762 0.8936],

Unexpected expenditures:

d = [0.0578; 0.3528; 0.8131; 0.0098; 0.1388],

Covering costs on quality evaluation (planned) in each quarter (four months):

A =[0.2027 0.2721 0.7467 0.4659; 0.1987 0.1988 0.4450 0.4186 ;0.6037 0.0152 0.9318 0.8462],

Notice: first column presents unit volume in adding up. Other accepts variability in reliability. The volume of finance on which the optimization requirement applies is:

b = [0.5251; 0.2026; 0.6721],

Limitations:

*lb = -0.1*ones(4,1),ub = 2*ones(4,1),*

Determination of optimal value *X* at lb=ub=0:

X=lsqlin(C,d,A,b),

Applied to conditional equation:

*q=(C*X-d),*

Q=q. ^2

Other two lines of matrix *Q* can be ignored. Then it is actually true that:

Q<_b.

At the acceptance of *lb* and *ub*:

X=lsqlin(C,d,A,b,[ ],[ ],lb,ub), presents costs optimization on maintenance of ground airport navigation complex reliability.

*q=(C*X-d),*

Q=q. ^2,

The result shows increased "density" of optimal control, however, the requirement of the equation is fulfilled [9].

Further evaluations lean of these conclusions:

- Control of environmental safety at a particular airport operator is recent and possible.
- It is possible to decrease the task complexity apart from the elements of capital inputs (matrix *C*).
- Quadratic criterion is purely selective.
- It is possible to evaluate environmental risk by inserting capital inputs into dominant element.
- Successful task solution requires quality input data [8].

V. Conclusion

Used statistic method shows the possibility of its use in practice to observe ergatic base complex and in connection with spent financial means to maintain its high reliability. The acceptance of ergatic base complex reliability effectiveness is highlighted by accidentally, which needs to be covered by planned complementary financial means [9]. Method of the estimation of maximum probability has loosely built-up structure, which accepts recent state of decreased reliability by prognosticated number of performed tests. Effectiveness estimation is observed by the criterion of dispersion. This method enables graphical and analytical record of sequence and further evaluation with help of statistic methods, see Fig.1. The method of costs based on the principle of quadratic evaluation enables *a priori (lat.)* to observe the development of ergatic base complex effectiveness. The method can be used by the operators of complex land aviation ergatic complexes in connection with the optimization of financial criterion of its operation.

Acknowledgment

This paper is published as one of the scientific outputs of the project: Centre of Excellence for Air Transport “ITMS 26220120065”.

References

[1] LAZAR, T, MADARASZ,L, GASPAR,V.: Process analysis estimating the efficiency of identifying MPM with intelligent control / Procesná analýza odhadu efektívnosti identifikácie MPM s inteligentným riadením (in Slovak), Elfa: Kosice , 2013, ISBN 978-80-8086-200-8

[2] NOVAK, A., MRAZOVA, M.: Research of Physiological Factors Affecting Pilot Performance in Flight Simulation Training Device, Communications - Scientific Letters of the University of Zilina, vol. 17, No. 3, 2015, 103-107, ISSN 1335-4205

[3] MADARASZ, L: Intelligent technologies and their applications in complex systems/ Inteligentné technológie a ich aplikácie v zložitých systémoch (in Slovak) University Press Elfa: Kosice, 2004, pp. 348, ISBN 80-89066-75-5.

[4] ZITEK, P.: Simulation of dynamic systems/ Simulace dynamických systemů (in Czech), SNTL - Nakladatelstvi technicke literatury : Praha, 1990, ISBN 80-03-00330-X, pp.147.

[5] TARAN, V. A.: Ergatic operation systems./ Ergatičeskije sistemy upravlenia (in Russian), Moskva : Masinostrojenie. 1976, BBK-kod O52-047-021.07, pp. 188.

[6] MATAS, M., NOVAK, A.: Models of Processes as Components of Air Passenger Flow Model, Communications : Scientific Letters of the University of Zilina, vol. 10, No. 2, 2008, 50-54, ISSN 1335-4205.

[7] LAZAR, T., BREDA, R., KURDEL, P.: Instrumental flight safety fuse. Technical University of Kosice / Inštrumentálne istenie letovej bezpečnosti.Technická univerzita v Košiciach (in Slovak), 2011, ISBN 978-80-553-0655-1. , pp. 33-41

[8] LAZAR, T, ADAMCIK, F., LABUN, J.: Feature Modeling and control of aircraft /Modelovanie vlastností a riadenia lietadiel (in Slovak), Technicka univerzita v Kosiciach, 2007, 317p., ISBN 978-80-8073-839-6.

[9] KURDEL, P., NOVÁK SEDLÁČKOVÁ, A., MREKAJ, B.: Hodnotenie ekonomickej efektívnosti a spoľahlivosti ergatického bázového komplexu. In: Perner´s Contacts [elektronický zdroj]. - ISSN 1801-674X. - Roč. 12, č. 1 (2017), online, s. 131-139.al Writer’s Handbook. Mill Valley, CA: University Science, 1989.

Supercombinator Driven Grammar Reconstruction

Michal Sičák, Ján Kollár
Technical University of Košice, Department of Computers and Informatics
Letná 9, 042 01 Košice, Slovakia
Email: {michal.sicak, jan.kollar}@tuke.sk

***Abstract*—In this paper we discuss the possibility of supercombinator driven grammar reconstruction. We present the way how to reconstruct a CFG to its original form from the supercombinator set. We also show that we can achieve mild form of grammar refactoring while performing the process. Additionally, we process multiple grammars into a single supercombinator set and from it we reconstruct the grammars back. We choose most used supercombinators with the arity of 10 to perform the reconstruction. We then analyze the reconstructed sentence parts.**

***Index Terms*—supercombinators, formal grammar, grammar reconstruction**

I. Introduction

Grammars are a useful mechanism to convey and abstract a language. The primary task of a grammar is to describe any language in a concise form. But as Klint et al. pointed out in [1], we can use grammars outside their primary scope. According to them, grammars are already widely used in software engineering. We have adopted this philosophy and we are using grammars as a prime object in our research. Our initial experiments showed that we can create a non redundant form and therefore battle phenomena called structural explosion [2]. Further, we have developed a transformation algorithm that can convert any Context-free grammar (CFG) into a non redundant set of supercombinators [3].

Supercombinator is a term coined by Hughes in [4], and it means a combinator that is composed only of other supercombinators or constants, therefore there are no free variables on any level inside of any supercombinator. Supercombinators are just pure lambda expressions. We are transforming grammars into these lambda calculus expressions, where we enhanced the calculus' definition with grammar operations that are occurring in the input grammars. For example, let's have a simple grammar excerpt shown in (1). It is just a simple alternative between two unspecified rule parts α and β. For the example purposes, we do not care right now, what those parts mean, we just need to know that they are valid parts of a CFG. Grammar operator between them is | - an alternative sign. By using our transformation algorithm, our grammar (1) is depicted as lambda expression in (2). We see that both α and β have been transformed into another supercombinator, depicted as L^α and L^β respectively. The argument part here is rather unknown so we assume that they both have exactly one argument, which in other words means a terminal symbol. We thus see the abstraction of the structure from the data. By using the basic composition, we can represent any CFG in such a form.

$$A \rightarrow \alpha \mid \beta \tag{1}$$

$$L^A = \lambda x_1.\ \lambda x_2.\ L^\alpha\ x_1 \mid L^\beta\ x_2 \tag{2}$$

We have used this algorithm to decompose various forms of CFGs into a single non-redundant supercombinator form. We have already used book samples processed to a CFG with the use of Sequitur algorithm [5], where we have shown rather significant reduction of grammar elements, the results are in our paper [3]. Further on, we have used the Groningen Meaning Bank (GMB) [6] that contains newspaper articles processed with Combinatory Categorial Grammar as our source. We were exploring the properties of supercombinator set in the paper [7]. We've found out that this set when given grammar excerpts of limited size creates a form that is size-bound by mathematical sequence called the Catalan number.

Although the supercombinator form represents the input grammars without loss of data, we have never published results where we try to reconstruct the original data back. Naturally, should we apply right arguments to the lambda expression, we would obtain the original grammar back. The grammar might be slightly changed in its appearance but not in the function, this depends on the properties of the original grammar, specifically whether it contains cycles or not. By using simple descriptive CFGs, like the ones that are the results of the Sequitur algorithm, we can obtain the original input sentences at the input, as the grammars in that case do not generalize.

The main contributions of this paper are:

- We present the reverse process, where we use our obtained supercombinators and by application of arguments, which are just words of sentences, we obtain the original grammar form back. We also note that we might achieve mild level of grammar refactoring. We present the reverse process in the section IV.
- We select semi-randomly most used supercombinators with the arity of 10 as our source for experiments, where we have a hypothesis that larger supercombinators that are frequently used represent subsentences of the input sample sentences.
- We present experimental results that confirm our hypothesis in the section VI. We evaluate the results that we've got from the supercombinator set obtained from GMB set abstraction.

This work was supported by project KEGA 047TUKE-4/2016 ”Integrating software processes into the teaching of programming”.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

II. Motivation

The main motivation behind the grammar reconstruction from supercombinator form is a process called grammar inference. It is a process of grammar acquisition from a set of samples. We know that it is impossible to infer grammar from positive samples purely algorithmically. This has been proven by Gold in [8]. We can partially overcome this problem with the use of heuristic or statistical methods. We are not trying to create a new method for grammar inference in this paper as our algorithm needs grammars on its input in the first place. In fact, we relay on that process. But eventually we might be able to infer some kinds of grammar forms that have not been previously present in the processed set.

We also lightly touch the field of natural language processing. We can find many approaches to parse sentences of natural language, for example Cambria and White offer a review of a state of the art in this field in [9]. Grammar inference is also researched question in the field of software engineering. Automated grammar acquisition seems to be rather interesting field as Stevenson and Cordy show in their review paper [10].

We work with grammars, which means that we need them on the input of our algorithm. The past results of our research show a promising guidance to our research. In our research [11] we have pointed out the phenomena of structural explosion. This phenomena is bound to grammars. It occurs for example when we are creating finite state automata from regular expressions. Let's have an example expression (3).

$$a\,b? \mid c\,d? \mid e\,f? \mid g\,h? \tag{3}$$

We can see that this expression is an alternative of four parts. Each of these parts has the same structure, but the terminal symbols are different. This means that should we convert this expression to automaton, we would get similar structures but in different strands. Let's have a look. On the Fig. 1 we see that all four strands of the expression have the same structure. We cannot generalize those strands just right away, since the regular expression $(a \mid c \mid e \mid g)\ (b? \mid d? \mid f? \mid h?)$ is more general version that accepts elements not acceptable with the original one. The solution to this problem is a supercombinator driven abstraction of grammars. We will explain it in the following section. Here we just mention that the part of the expression $x\ |y?$ can be abstracted as a single supercombinator that is then stored in the resulting form only once.

III. Brief Description of Transformation Process

We have mentioned that with the use of our process, we can abstract grammars into more compact form that prevents structural explosion. Example shown in the previous section deals with regular expressions, which our algorithm can abstract. But it can do more, it can abstract context-free grammars as well. We will briefly describe this process in this section. For more detailed explanation, please see our other work [7].

The best way to explain the transformation process is by examples. Let's have a simple CFG depicted as rules (4) and (5). This grammar consists of two non-terminal symbols A and B and the same number of terminal symbols, namely a and b. It contains two grammar operations, sequencing and alternative. This translates to our supercombinator form. As supercombinators are lambda expressions, we need to extend basic definition of lambda calculus with the exact amount of grammar operations that we are going to use. In this case they are just two.

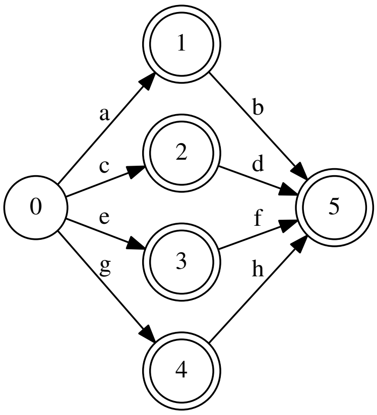


Fig. 1: Example of an automaton suffering from the structural explosion.

$$A \rightarrow a\ B\ a \mid a \tag{4}$$

$$B \rightarrow b \mid A \tag{5}$$

Each supercombinator is created with only one grammar operation in its body. Needles to say, the alternative has lower priority than sequencing. Which means that a supercombinator, which is build around alternative operation will contain supercombinators with sequencing operation in its body.

The final resulting form is non-redundant, which means that no two equal supercombinators are present there. And it also means that we can merge smaller sets of supercombinators together as it is indeed a one set of supercombinators that we are creating. We can thus process each grammar rule separately and then merge the resulting sets together. This cannot be done right away, as we can see that both rules refer to each other. To prevent the infinite cycle, the first step of our transformation is a construction of graph from the grammar. From that graph we can obtain argument strings for each non-terminal. Argument strings are obtained by using depth first search of this graph starting from our non-terminal and gathering only terminals. For example, argument string for $A\ =\ a\ b\ a\ a$. Should we want to remove duplicates, and we can, then the argument string for $A\ =\ a\ b$[1]. Now we know the arity of our future L^A supercombinator, it is 4. We can construct small sets from each rule now, since the non-terminal jump can be replaced by temporary supercombinator with the identified arity and later, when the actual supercombinator is created, we can replace the temporary one.

[1] By not removing the duplicates we might obtain more compact form, but this is not necessarily true for all grammar types. See [7] for more information about this problem.

TABLE I: Supercombinator form of the grammar in (4) and (5).

Name	Supercombinator Body	Arguments
L^0	$\lambda x_1.x_1$	$\{a, b\}$
L^{A_1}	$\lambda x_1.\lambda x_2.\ L^{A_2}\ x_1\ x_2 \mid L^0\ x_1$	$\{a\,b\}$
L^{A_2}	$\lambda x_1.\lambda x_2.L^0\ x_1\ +\ L^B\ x_2\ x_1\ +\ L^0\ x_1$	$\{a\,b\}$
L^B	$\lambda x_1.\lambda x_2.\ L^0\ x_1 \mid L^{A_1}\ x_2\ x_1$	$\{b\,a\}$

Each rule is split by its operators to create a supercombinator set, which means that rule A will create 3 supercombinators and rule B two. Each time the identity combinator is created, therefore we have one more supercombinator per number of grammar operations. Created sets are then merged, thus creating final set of supercombinators. The set created from our example is in the Table I[2]. In this table we can see that four supercombinators have been created in total. The arguments in the last column represent the possible arguments for that supercombinator. They are stored separately in our form, but by their application we can obtain the entire grammar back in almost ordinary lambda beta reduction fashion. We will explain this in the following section.

IV. Grammar Reconstruction

We have shown the outline of transformation algorithm that transforms grammars into a set of supercombinators. The set fully represents the input grammar or grammars, as it can contain more than one. In order to obtain the original grammar back we just need to apply arguments of the grammar to the top supercombinator. Top supercombinator is the largest supercombinator of grammar that represents starting non-terminal symbol of that grammar.

Ordinary beta-reduction of top supercombinators with its arguments is possible in case the grammar does not contain cycles. With cycles, should we perform ordinary beta-reduction, we would end up in the endless cycle. Let's show what we mean on an example. See, what happens if we apply arguments on the top supercombinator of our grammar represented by rules (4) and (5) in the reduction process depicted in (6). We see that this reduction would go indefinitely.

$$\begin{aligned}
&L^{A_1}\ a\ b\\
&\to\ \lambda x_1.\lambda x_2.\ L^{A_2}\ x_1\ x_2 \mid L^0\ x_1\ a\ b\\
&\to\ L^{A_2}\ a\ b \mid L^0\ a\\
&\to\ \lambda x_1.\lambda x_2.L^0\ x_1\ +\ L^B\ x_2\ x_1\ +\ L^0\ x_1\ a\ b \mid a\\
&\to\ a\ +\ (\lambda x_1.\lambda x_2.\ L^0\ x_1 \mid L^{A_1}\ x_2\ x_1)\,a\,b + a \mid a\\
&\to\ \dots
\end{aligned} \tag{6}$$

However, should use labels to represent at least some non-terminals, we can terminate this application process. For example, we see in the last line of reduction (6) that we are trying to reduce supercombinator L^{A_1} for the second time. Here we need to swap it with the label that points to that particular rule, which is the rule A. See what happens in (7). We see that this is virtually identical grammar to the one in rules (4) and (5). We can see that some grammar refactoring happened here. Yet in case we use label also for the other non-terminal B, we obtain the original grammar back.

$$L^{A_1}\ a\ b\ \to\ a\ +\ (b \mid A) + a \mid a \tag{7}$$

However as already mentioned, this is needed only in the case if the grammar contains cycles. Should we use a form of a simple CFG or a regular expression, this labeling is not needed and just by performing standard beta-reduction, we obtain the original grammar back.

What happens in case we have more grammars in our set? As you can see in the Table I, we show arguments for each supercombinator. They are sets of possible argument strings for each supercombinator, for example L^0 has two strings, namely string a and string b. Other supercombinators have only one string in their set. In order to perform beta-reduction we just need to remember, which string in the set of top supercombinator is the one that recreates the original grammar back. Remember, our set is non redundant, and top supercombinator of one grammar might be just a part of another grammar.

Fortunately, we use only noncyclic grammars in our experiments presented in this paper, so the reconstruction is rather straightforward.

V. Grammar Samples

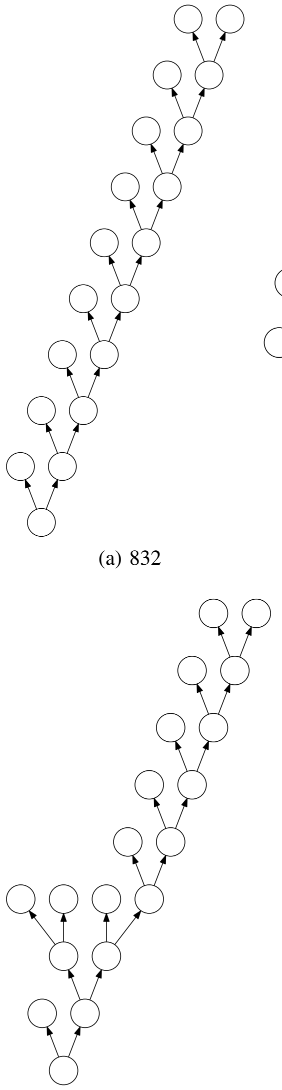
We have processed GMB corpus with our algorithm in order to obtain large supercombinator set. We have explored properties and features of supercombinator set in the paper [7]. We will work with the same set here, but with the different aim this time. However, to make this paper self contained, we are going to briefly describe our input set in this section.

GMB is a corpus that consists of 10 000 processed short newspaper articles. They are processed with the combinatory categorial grammar (CCG) [12]. The corpus itself can be used to search for meaning and other semantic purposes that natural language processing offers. We have chosen to strip almost all semantic information away and extract only the plain structure of parsed sentences. These structures can be described in the form of CFG. And we can process those into one non-redundant set, which we have done.

We have transformed totally 62 008 sentences into equal number of grammars. And they have been processed into a set of 246 315 supercombinators. Only about 25% of supercombinators that have been created were unique. The rest have occurred more than once, where about 50% of all initially created supercombinators were merged more than 1 000 times. This means that we have a lot of structures reoccurring in the original set. Some possible supercombinators were never created. This is more or less obvious result, if we consider that a natural language does not contain random structures.

How those supercombinators look like then? We have only binary supercombinators in our set, which means that each

[2] Note that to represent the concatenation operation we are using plus (+) sign, as the empty space or a dot are already parts of the standard lambda calculus.



(a) 832

(b) 580

(c) 576

(d) 523

Fig. 2: Most used supercombinators with the arity of 10. We can see how many original grammars contained them bellow the figures themselves.

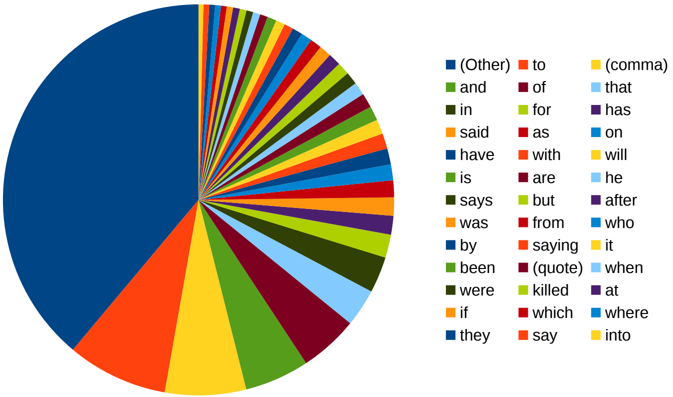


Fig. 3: Cake diagram that shows the count of starting words of sentence parts obtained from the top four supercombinators with the arity of 10. The values are in counterclockwise order, starting from the largest part (Other).

supercombinator (apart from the L^0) has a grammar operation that is binary. All grammar operations are concatenations. Therefore we can depict supercombinators as a binary trees. Remember that all structural information i.e which word is noun, or which word is verb has been stripped away, we are only using plain hierarchy to differentiate between supercombinators.

We are going to use supercombinators with the arity of 10 in our experiments. Only the most frequent supercombinators are going to be used, we see four most used supercombinators with the arity of 10 in the Fig. 2.

The leaf nodes represent the L^0 supercombinator, where each level represents supercombinators as all other supercombinators are binary. We see that all four of them eventually contain the same supercombinator with the arity of 5. This similarity might indicate similarity of sentences they represent. We can see that these four represent totally 2 511 sentences from the original input set. As this is a rather large number we need to employ the power of visualization to grasp the idea what those supercombinators represent. We see that they are quite similar supercombinators, so we are going to analyze their word strings together in the next section.

VI. Analysis of Selected Sentence Sections

When we take the four most occurring supercombinators with the arity of 10 and apply on them their argument strings, we get parts of the sentence they represent in the original source material. In the Fig 3 we see the most used starting words of these sentence parts. Starting words are a good indicators of what those sentence parts represent. It could mean that structures are rather randomly distributed should we get a lot of different starting words. But we argue, and the results seem to confirm our hypothesis that those sentence parts are subsentences of the whole sentences that are the original source material.

We have put all words with the count less than 10 in the same category that we've named (Other). We see that majority of the share is taken by the **other** category, yet there are some words that occur rather frequently. Most frequent word is **to**, the second place belongs to the comma (**,**), which is valid CCG element. We also have another punctuation mark in our diagram, and that is a quote (**"**). We see that all top initial words belong to conjunctions. This seems to indicate that even though we have chosen supercombinators with the arity of 10 rather randomly, those strings are subsentences. This is probably due to the fact that we have selected the most occurring and long enough structures. Thus, we have found meaningful sentence parts, even though we are using sentence structures cleared of all initial CCG meaning.

We also provide Table. II to show the exact number of occurrence of these words. Among the conjunctions that we have found we have also some variations of **to be** and **to have**

TABLE II: Starting words and the amount of their occurrence in the reconstructed sentence parts.

Word	Count	Word	Count	Word	Count
(Other)	978	with	32	it	19
to	209	will	30	been	19
(comma)	168	is	30	(quote)	16
and	134	are	30	when	15
of	122	he	27	were	14
that	78	says	26	killed	14
in	75	but	26	at	14
for	48	after	26	if	13
has	39	was	25	which	12
said	38	from	24	where	12
as	35	who	23	they	12
on	33	by	21	say	12
have	33	saying	19	into	10

verbs as well as the personal pronouns. As these are newspaper articles it is no wonder that we have three forms of the verb **to say** in there. And also the verb **to kill** just underlines that fact as well.

VII. Discussion

The result presented in the previous section were obtained just from the four most occurring supercombinators with the arity of 10. The randomness of selection of this specific arity seems not to be a problem and our results seem to indicate that large most occurring supercombinators that have been obtained from sentences represent meaningful subsentences. Even though we have only used pure structure of the sentences, abandoning all concepts that CCG uses like what is a verb and what is a noun.

The properties of supercombinators themselves are not interesting for the purpose of this paper, as they have been already researched in our previous paper [7]. This paper's aim was to present more practical results of our supercombinator approach. Here we show that by dissolving grammars into one supercombinator set we can eventually obtain meaningful results.

One other interesting bit that we have found is that this method opens a possibility of grammar inference from the sentences. Subsentences starting with the same words might belong to the same category and thus we can generalize the original CFGs without the fear of over-generalization and create some meaningful grammatical representation. This is however just an idea for further research and is not a goal of this paper.

VIII. Conclusion

We have presented the ability of grammar reconstruction in this paper, both in the theoretical abstract way on the pure lambda calculus and also on the level of natural language sentences. The reconstructed sentence parts of semi-randomly selected supercombinators show signs of meaning even though the supercombinators were constructed from pure structural information.

Presented results show that we can reconstruct original grammars that have been dissolved into supercombinator set and we can even think about some form of an automated grammar inference or refactorization.

References

[1] P. Klint, R. Lämmel, and C. Verhoef, "Toward an engineering discipline for grammarware," *ACM Trans. Softw. Eng. Methodol.*, vol. 14, no. 3, pp. 331–380, Jul. 2005. doi: 10.1145/1072997.1073000. [Online]. Available: http://doi.acm.org/10.1145/1072997.1073000

[2] J. Kollár, M. Sičák, and M. Spišiak, "Towards machine mind evolution," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015. doi: 10.15439/2015F210 pp. 985–990. [Online]. Available: http://dx.doi.org/10.15439/2015F210

[3] M. Sičák and J. Kollár, "Supercombinator set construction from a context-free representation of text," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 8. IEEE, 2016. doi: 10.15439/2016F334 pp. 503–512. [Online]. Available: http://dx.doi.org/10.15439/2016F334

[4] R. J. M. Hughes, "Super-combinators a new implementation method for applicative languages," in *Proceedings of the 1982 ACM symposium on LISP and functional programming*. ACM, 1982. doi: 10.1145/800068.802129 pp. 1–10. [Online]. Available: http://dx.doi.org/10.1145/800068.802129

[5] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical strcture in sequences: A linear-time algorithm," *J. Artif. Intell. Res.(JAIR)*, vol. 7, pp. 67–82, 1997. doi: 10.1613/jair.374. [Online]. Available: http://dx.doi.org/10.1613/jair.374

[6] V. Basile, J. Bos, K. Evang, and N. Venhuizen, "Developing a large semantically annotated corpus," in *LREC 2012, Eighth International Conference on Language Resources and Evaluation*, 2012. [Online]. Available: https://hal.inria.fr/hal-01389432

[7] M. Sičák and J. Kollár, "Properties and limits of supercombinator set acquired from context-free grammar samples," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 11. IEEE, 2017. doi: 10.15439/2017F249 pp. 711–720. [Online]. Available: http://dx.doi.org/10.15439/2017F249

[8] E. M. Gold, "Language identification in the limit," *Information and control*, vol. 10, no. 5, pp. 447–474, 1967. doi: 10.1016/S0019-9958(67)91165-5. [Online]. Available: http://dx.doi.org/10.1016/S0019-9958(67)91165-5

[9] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research," *IEEE Computational intelligence magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[10] A. Stevenson and J. R. Cordy, "Grammatical inference in software engineering: an overview of the state of the art," in *Software Language Engineering*. Springer, 2013, pp. 204–223.

[11] J. Kollár, M. Spišiak, and M. Sičák, "Abstract language of the machine mind," *Acta Electrotechnica et Informatica*, vol. 15, no. 3, pp. 24–31, 2015. doi: 10.15546/aeei-2015-0025. [Online]. Available: http://dx.doi.org/10.15546/aeei-2015-0025

[12] M. Steedman and J. Baldridge, "Combinatory categorial grammar," *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell*, 2011.

Adapted parallel Quine-McCluskey algorithm using GPGPU

Vladimír Siládi
Faculty of Natural Sciences
Matej Bel University
Banská Bystrica, Slovakia
vladimir.siladi@umb.sk

Michal Povinský
Faculty of Natural Sciences
Matej Bel University
Banská Bystrica, Slovakia
michal.povinsky@umb.sk

L'udovít Trajtel'
Faculty of Natural Sciences
Matej Bel University
Banská Bystrica, Slovakia
ludovit.trajtel@umb.sk

Maxatbek Satymbekov
Institute of Information systems
Kazakh National University named after Al-Farabi
Almaty, Kazakhstan
m.n.satymbekov@gmail.com

***Abstract*—This paper deals with parallelization of the Quine-McCluskey algorithm. This algorithm is a method used for minimization of boolean functions. The algorithm has a limitation when dealing with more than four variables. The problem computed by this algorithm is NP-hard and run-time of the algorithm grows exponentially with the number of variables. It is possible to adapt the Quine-McCluskey algorithm for running on a parallel computing system. The graphics processing units (GPUs) brings acceleration of computing process. This solution differs from the previous solution in the use of bit fields. Parallelization of the algorithm is implemented through Compute Unified Device Architecture (CUDA).**

***Index Terms*—Graphics processing unit, Minimization methods, Digital circuits, Boolean functions, Hardware**

I. Introduction

English mathematician and philosopher G. Boole invented his algebra in 1854 [1]. A few decades later C. E. Shannon showed how the *Boolean algebra* can be used in the design of digital logic circuits [21]. A boolean function is a function that produces a boolean value output by logical calculation of boolean inputs. It plays key roles in programming algorithms and design of digital logic circuits. Minimization of boolean function (using Boolean laws) is able to optimize the algorithms and digital logic circuits. Finding and minimization of boolean functions (normal forms) is one of the crucial problems in logic circuits design.

Two popular techniques for minimization/simplification of boolean functions are used:

- Karnaugh map (the method based on a graphical representation of boolean functions) [12],
- the Quine-McCluskey algorithm (the Method of Prime Implicants; sometimes referred to as the Tabulation Method) [16].

Both of these methods are mechanical in nature (their efficiency depend on the designer's abilities). The methods designed for computer and human (e.g. Quine-McCluskey method, Karnaugh, Svoboda, Veitch, Marquand maps,...) has got advantages and disadvantages. We can find some comparison of them [17]. Karnaugh map is a diagrammatic and mechanical technique based on a special form of Venn diagram [24]. Usually it is used to handle boolean functions (expressions) with no more than six variables. But, Karnaugh map could be used for another purpose e.g. artificial intelligence to map human mind, high performance computing to build networking logics [15].

The Quine-McCluskey algorithm is a method used for minimization of boolean functions. The method was formulated by Quine and later improved by McCluskey [19] [20] [16]. The Quine-McCluskey method is a tabulation method. It is functionally identical to Karnaugh Map [12], but this method makes it more efficient (the tabular form overcomes the limitation associated with graphical representation) for use in computer algorithms (method is *Computer Based Technique* for minimization of boolean functions; it gives a deterministic way to check that the minimal form of a boolean function has been reached).

The method involves two steps:

1) determination of prime implicants (finding all prime implicants of the function),
2) selection of essential prime implicants (use those prime implicants in a prime implicant chart to find the essential prime implicants of the function, as well as other prime implicants that are necessary to cover the function).

The combining minterms with other minterms as a part of the Quine-McCluskey algorithm is natural parallel. In this paper we present an improved parallel algorithm for the first step of the Quine-McCluskey method.

II. Related work

The Quine-McCluskey method, a tool of discrete mathematics, is very simple and systematic technique for minimizing boolean functions [11]. However, the computational complexity of the Quine-McCluskey algorithm is $O\left(N^{\log_2 3}\log_2 N\right)$, with input length $N = 2^n$ [23]. The minimization of a boolean function is (probably) a hard

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

problem [23] [25]. Prasad, Beg and Singh analysed the behavior of Quine-McCluskey simplification for different number of product terms and introduced a mathematical model to predict the boolean space complexity. They mathematically modelled boolean function complexity by the following equation:

$$N = \alpha \cdot t^{\beta} \cdot e^{-t \cdot \gamma} + 1 \tag{1}$$

where,

N is the number of literals,

t is the number of non-repeating product terms in the boolean function,

α, β and γ are three constants depend on the number of variables [18]. A couple of solutions were introduced to quickly and automatically simplify boolean functions [9] [10] [11]. Majumder et al. designed a technique based on decimal values. This technique decreases the probability of an error occurrence [13]. Non-deterministic, stochastic and heuristic computational models offer interesting tools to find unconventional alternative solutions [7] [8]. El-Bakry et al. used neural networks [2], fast neural networks [3], modular neural networks [4] for the minimization (simplification).

Another question of this minimization problem is the choice of a suitable data structure for representing boolean functions and an associated set of manipulation algorithms. Graph-based solution have shown Bryant [6].

In our previous work [22] we presented an parallel implementation of Quine-McCluskey algorithm on graphics processing unit through Compute Unified Device Architecture (CUDA). We designed parallel function, which naive transcribed natural parallelism of the Quine-McCluskey algorithm. The parallel implementation utilised a memory inefficiently.

III. Novel algorithm for simplification

We process the data in multiple step. Basically, the original Quine-McCluskey algorithm is adopted. New approach differs in terms representation. The goal of this algorithm is more intensive memory usage. A bitmap representation is used.

In each step of the algorithm, implicants are first partitioned by the positions of dashes in the terms. Each partition is then scanned for mergeable terms. Small partitions use a simple $O\left(N^2\right)$ algorithm, while large partitions use our $O\left(N\right)$ algorithm.

In our term merging algorithm the list of terms is first transformed into a bitmap (bit array in term of programming language C), representing each term as one set bit. Dashes in the term are treated as zeroes when calculating the bit's index, for example:

$$\begin{array}{ccccc} (1-10-0) & \longmapsto & 101000 & \longmapsto & (40)_{10} \\ \mathit{minterm} & & \mathit{array\ \ of\ \ bits} & & \mathit{index.} \end{array}$$

Then the list of terms is scanned and for each term all suitable terms are looked up in the bitmap to merge with. Two conditions are applied:

- the current term differs from the term to merge with in exactly one bit,
- the different bit is zero in the current term.

This means only a small number of positions in the bitmap are checked for each term. The partitioning ensures the correct merging (dashes are represented with zeros). Duplicate terms are also never generated. The merged rows are then written to a new list to be used by the next step of the algorithm and both original terms are marked as used. If both of the original rows were optional, the new row will also be marked as optional.

IV. CUDA implementation

The Quine–McCluskey algorithm has got naturally parallel parts, which can be implemented on the SIMD architecture and therefore on the GPU (SIMT architecture). Our term merging algorithm contains this parallel part. Therefore, another significant improvement is expected to achieve a speed up of computation by implementation on a GPU. The design of the sequential CPU implementation was based on the parallel GPU implementation (*see* Listing 1). Listing 1 shows the GPU kernel used for finding merge-able implicants in a partition.

Listing 1. Part of the parallel implementation on GPU

```
 1  static __global__ void bitarray(int *vars, int bits,
 2   int dashes, int *bitarray,
 3   uint32_t *used, uint32_t *output)
 4  {
 5    .
 6    .
 7    .
 8   if (row_in_block == 0 &&
 9    bit_in_row < ROWS_PER_BLOCK) {
10    rows[bit_in_row] = bitarray[rownumber+bit_in_row];
11    usedbuf[bit_in_row] = 0;
12   }
13
14   __syncthreads();
15   found[bit_in_block] = 0;
16   if (rows[row_in_block] & (1<<bit_in_row)) {
17    for (j=0; j<bits; j++) {
18     uint32_t m = 1 << j;
19     if (dashes & m) continue;
20     uint32_t otheridx = bitnumber^m;
21     if (!(otheri & (1<<(otheridx%32)))) continue;
22     atomicOr(usedbuf+row_in_block, (1<<bit_in_row));
23     if (bitnumber & m) continue;
24     if ((m-1) & dashes) continue;
25     found[bit_in_block] |= m;
26    }
27   }
```

First of all, the bitmap is loaded into shared memory and flag of use is set on zero for each representation of minterm (rows 7–9). After threads synchronisation, the combining minterms with other minterms is executed by commands on rows 14–24. These parts of the program code are implementation of above mentioned natural parallelism by CUDA. The kernel runs on GPU (device) and other parts of the algorithm run on CPU (host). This requires the transfer of data from the host to the device and vice versa.

The kernel is called with the following arguments:

- status variable array (`int *vars`),
- term size in bits (`int bits`),

- positions of dashes in the current partition (`int dashes`),
- term bitmap (`int *bitarray`),
- used row flag array (`uint32_t *used`),
- output array (`uint32_t *output`).

Each block processes a number of terms. Each group 32 threads processes 32 terms, each thread looking up a term differing in a bit at one position. The results are then written to an output array.

V. Experiments and results

The number of bits in results is the number of inputs to each function, so the number of output values for each function is $2^{number\ of\ bits}$. Each optimized function was randomly generated and the value of fill is the probability of each bit being set to 1. Higher fill values cause more terms to be created, which significantly increases both time and memory requirements, *see* Fig. 1.

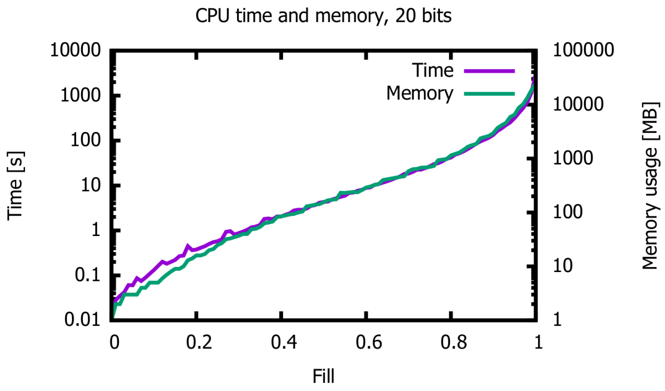


Fig. 1. CPU time and memory usage for a 20 bit function for different fill values

This limited our experiments to either small functions or low fill values. We were experimenting with functions of 20 input variables (*see* Fig. 2), 24 input variables (*see* Fig. 3), 28 input variables (*see* Fig. 4), 32 input variables(*see* Fig. 5).

For experiments on the GPU, three computers each with 64GB RAM, two Xeon E5-2650v2 CPUs and two Tesla K20m graphics cards with compute capability 3.5 (denoted *GPU1*), one computer with Geforce 9800GT GPU with compute capability only 1.1 (denoted *GPU2*) and one computer with Geforce GTX260M GPU with compute capability 1.3 (denoted *GPU3*) were used. For CPU experiments, computers with two Xeon X5670 CPUs and 48GB RAM and computers with two Xeon E5-2650v2 CPUs and 128GB RAM were used (in both cases denoted *CPU*).

For larger experiments, we were able to use on-disk storage for temporary implicant lists, *see* Fig. 5. Unfortunately we were unable to use this with GPU version of the algorithm. While for the smaller tested problem instances running on the GPU is slower than running on the CPU, we expect that GPU would be significantly faster for larger instances.

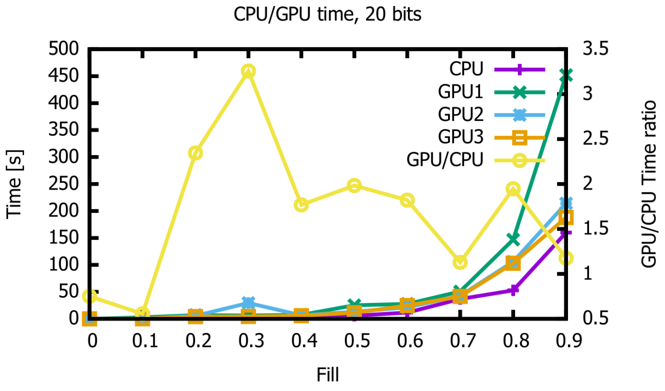


Fig. 2. CPU and GPU time for 20 bit functions

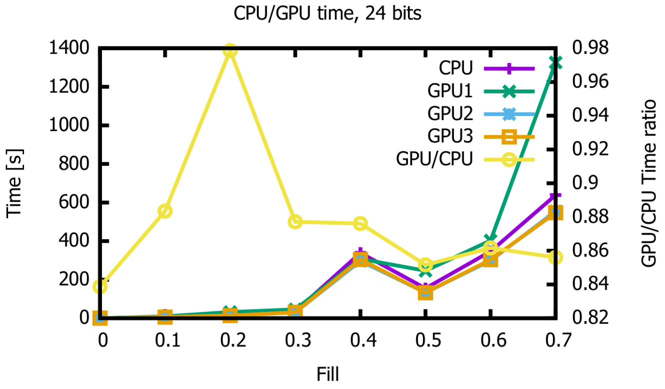


Fig. 3. CPU and GPU time for 24 bit functions

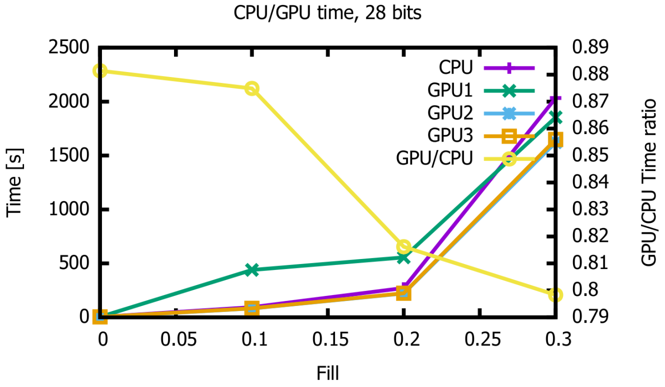


Fig. 4. CPU and GPU time for 28 bit functions

We were unable to test larger instances, as the host memory requirements grow exponentially.

Finally, one of the experiments ran on more powerful graphics card GeForce GTX 1080 Ti with computing capability 6.1 to prove weaknesses in the algorithm. We measured GPU memory usage for different problem sizes using nvidia-smi. GPU memory usage is constant during execution of the program, so no advanced memory profiling

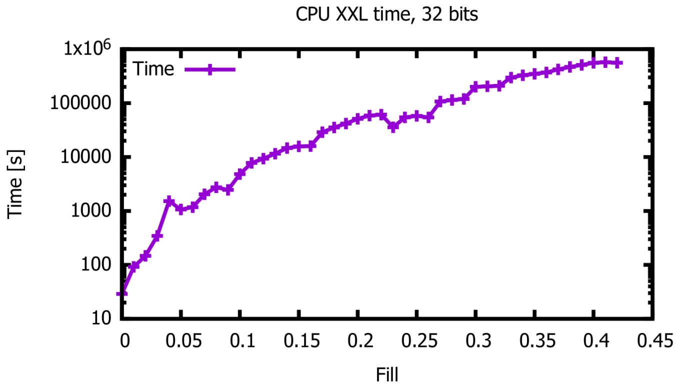


Fig. 5. CPU times for 32 bit functions with out-of-core storage

is required, *see* Fig. 7.

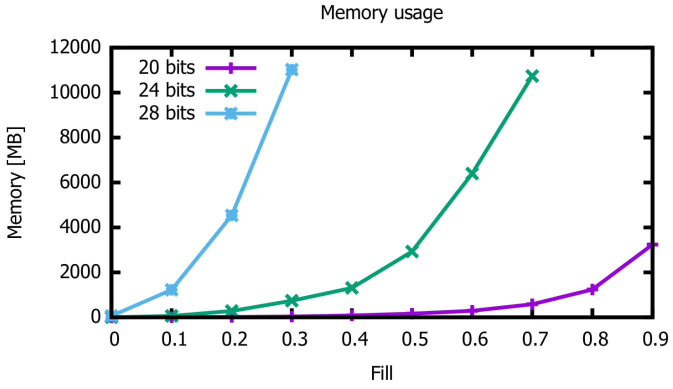


Fig. 6. On-disk storage using for temporary data – memory usage

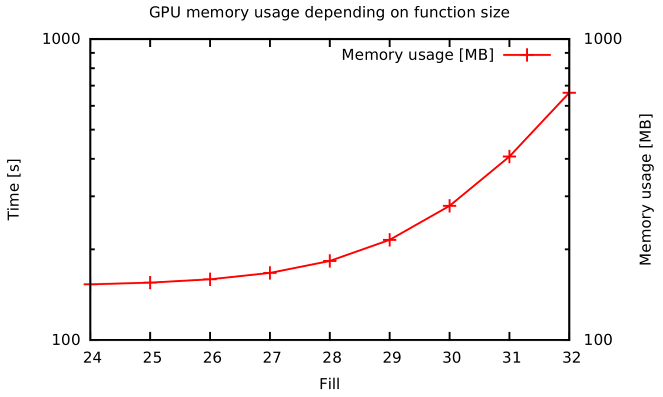


Fig. 7. GPU memory usage on GeForce GTX 1080 Ti depending on function size

Unfortunately, our improved algorithm has not saved memory of the host, but it saved memory of the device. In addition, the profiler conclusion is: "This kernel exhibits low compute throughput and memory bandwidth utilization relative to the peak performance of GeForce GTX 1080 Ti. These utilization levels indicate that the performance of the kernel is most likely limited by the latency of arithmetic or memory operations. Achieved compute throughput and/or memory bandwidth below 60% of peak typically indicates latency issues." (Fig. 8).

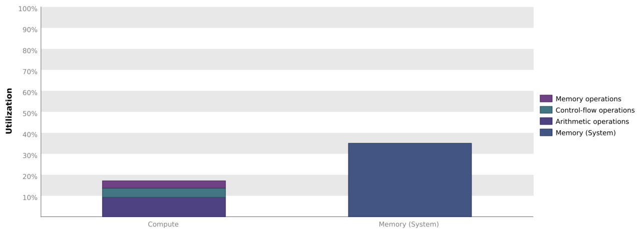


Fig. 8. Kernel performance by instruction and memory latency

The kernel's achieved occupancy of 48.1% is significantly lower than its theoretical occupancy of 100%. Most likely this indicates that there is an imbalance in how the kernel's blocks are executing on the shared memories so that all shared memories are not equally busy over the entire execution of the kernel.

The experiment has shown that this parallelization concept is still not very suitable for implementation on GPU.

VI. Conclusions

The CUDA implementation has been modified according to the new algorithm. Unfortunately, the algorithm still requires a large amount of CPU memory, which often limited the size of problems, which we could solve on our hardware. While for the smaller tested problem instances running on the GPU is slower than running on the CPU, we expect that GPU would be significantly faster for larger instances. We were unable to test larger instances, as the host memory requirements grow exponentially. Although presented parallel algorithm utilises the device memory more efficiently than our previously published parallel algorithm, it is still not very suitable for implementation on GPU. This way of adapting the natural parallel parts of the Quine-McCluskey algorithm on GPU still does not improve significantly the speed-up of the minimisation process. In future, research effort should be focused on optimisation of host's part of the algorithm. There is the weakness of this implementation on GPU.

Acknowledgment

Part of the computing was performed in the High Performance Computing Centre of the Matej Bel University in Banská Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

References

[1] G. Boole, “An Investigation of the laws of thought: On which are founded the mathematical theories of logic and probabilities,” Watchmaker Publishing, 2010.

[2] H. M. El-Bakry, “Fast Karnough map for simplification of complex boolean functions,” In: Proc. of 10th WSEAS International Conference on Applied Computer Science (ACS’10), Japan, pp. 478–483, 2010.

[3] H. M. El-Bakry, A. Atwan, “Simplification and implementation of boolean functions,” International Journal of Universal Computer Sciences, vol. 1, Issue 1, pp. 19–33, 2010.

[4] H. M. El-Bakry, N. Mastorakis, “A fast computerized method for automatic simplification of boolean functions,” In: Proceedings of the 9th WSEAS International Conference on Systems Theory and Scientific Computation (ISTASC ’09), Moscow, Russia, pp. 99–107, 2009.

[5] H. M. El-Bakry, A. Atwan, N. Mastorakis, “A new technique for realization of boolean functions,” In: Proc. of Recent Advances in Artificial Intelligence, Koweledge Engineering and Databases, Cambridge, UK, pp. 260–270, 2010.

[6] R. E. Bryant, “Graph-based algorithms for boolean function manipulation,” IEEE Transaction on Computers, vol. C-35, no. 8, pp. 677–691, 1986.

[7] I. Dirgová Luptáková, M. Šimon, L. Huraj, J. Pospíchal, “Neural gas clustering adapted for given size of clusters,” Mathematical Problems in Engineering, vol. 2016, Article ID 9324793, p. 7, 2016.

[8] I. Dirgová Luptáková, J. Pospíchal, “Maximum Traveling Salesman Problem by Adapted Neural Gas,” In: Advances in Intelligent Systems and Computing book series (AISC, volume 576), International Conference on Soft Computing-MENDEL. Springer, Cham, pp. 168–175, 2016.

[9] A. Duşa, A. Thiem, “Enhancing the minimization of boolean and multivalue output functions with eQMC,” The Journal of Mathematical Sociology, 39:2, pp. 92–108, 2015, DOI:10.1080/0022250X.2014.897949

[10] B. Gurunath, N.N. Biswas, “An algorithm for multiple output minimization,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 8, Issue 9, pp. 1007–1013, IEEE, 1989.

[11] T. K. Jain, D. S. Kushwaha, A. K. Misra, “Optimization of the Quine-McCluskey method for the minimization of the boolean expressions,” Fourth International Conference on Autonomic and Autonomous Systems (ICAS’08), Gosier, 2008, pp. 165–168.

[12] M. Karnaugh, “The map method for synthesis of combinational logic circuits,” Transactions of the American Institute of Electrical Engineers, vol. 72 part I,pp. 593–598, 1953.

[13] A. Majumder, B. Chowdhury, A. J. Mondai, K. Jain, “Investigation on Quine McCluskey method: A decimal manipulation based novel approach for the minimization of boolean function,” International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015, DOI: 10.1109/EDCAV.2015.7060531.

[14] V. Manojolovič, “Minimization of Switching Functions using Quine-McCluskey Method,” International Journal of Computer Application, vol. 82, no. 4, pp. 12–16, 2013.

[15] R. Mehta, R. Saini, N. Mudgal, M. Dhankar, “Delivering High Performance Result with Efficient Use of K-Map,” International Journal of Control and Automation, vol. 9, no. 2, pp. 307–312, 2016.

[16] E. J. McCluskey, “Minimization of boolean functions,” Bell System Technical Journal, vol. 35, Issue 6, pp. 1417–1444, 1956.

[17] M. Petrík, “Quine–McCluskey method for many-valued logical functions,” Soft Computing, vol. 12, Issue 4, pp. 393–402, Springer-Verlag, 2007.

[18] P. W. Chandana Prasad, Azam Beg, Ashutosh Kumar Singh, “Effect of Quine-McCluskey simplification on boolean space complexity,” In: Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2009, IEEE, 2009.

[19] W. V. Quine, “A way to simplify truth functions,” The American Mathematical Monthly, vol. 62, no. 9, pp. 627–631, Mathematical Association of America, 1955.

[20] W. V. Quine, “The problem of simplifying truth functions,” The American Mathematical Monthly, vol. 59, no. 8, pp. 521–531, Mathematical Association of America, 1952.

[21] C. E. Shannon, “A symbolic analysis of relay and switching circuit,” Electrical Engineering, vol. 57, Issue 12, pp. 713–723, IEEE, 1938.

[22] V. Siládi, T. Filo, “Quine-McCluskey algorithm on GPGPU,” In: AWERProcedia Information Technology and Computer Science, vol. 4 3rd World Conference on Innovation and Computer Science (INSODE-2013), pp. 814–820, 2013.

[23] S. P. Tomaszewski, I. U. Celik, G. E. Antoniou, “WWW-based boolean function minimization,” International Journal of Applied Mathematics and Computer Science,vol. 13, no. 4, pp. 577–583, 2003.

[24] J. Venn, “On the diagrammatic and mechanical representation of propositions and reasonings,” Philosophical Magazine, Series 5, 10:59, 1–18, DOI: 10.1080/14786448008626877.

[25] I. Wegener, “The complexity of boolean functions,” John Wiley & Sons, Inc. New York, NY, USA,1987.

Fully Automatic Modular Theorem Prover with Code Generation Support

František Silváši
Technical University Of Košice
Faculty of Electrical Engineering and Informatics
Email: frantisek.silvasi@tuke.sk

Martin Tomášek
Technical University Of Košice
Faculty of Electrical Engineering and Informatics
Email: martin.tomasek@tuke.sk

Abstract—**We present a theorem prover capable of autonomously proving and subsequently synthesizing executable implementations of parameterized theorems. The overall design is modular, allowing for changes in source and target language, underlying calculus of deduction and proof construction strategies. We provide an example based on System F, with a custom specification language, Haskell target language and a heuristic strategy for proof inference. We also introduce an intermediate language for proof script representation, making the process of proof reconstruction trivial. Results are verified by generating a subset of Haskell's standard Prelude library, demonstrating that even systems without very expressive theoretical underpinnings (such as System F, or similar calculi) are sufficient for rudimentary proof / code inference.**

I. Introduction

Theorem provers come in various flavours, often cleverly disguised as programming languages with particularly expressive type systems. There are many criterions by which they can be categorized and such choices are generally made by author(s) of any particular one. Just to name a few, underlying logical framework, the preferred way of writing proof scripts with regards to proof language and subsequent interpretation of proofs, and the amount of automation whether it is implicit or available on demand.

First we present a general modular approach to constructing a theorem prover that is independent of any of the aforementioned choices; this is done by utilizing an intermediate proof representation that differs from the standard notion of proof objects, which are generally functional terms in an internal language of any particular tool in question. Second, we (figuratively speaking) instantiate this template (while also constructing a working implementation thereof) with a framework based on System F, while also providing a set of fully automatic heuristic procedures for proof construction; those are then utilized to find proofs for propositions coinciding with a subset of function definitions from the Haskell's Prelude[9] library, which can be viewed as a standard library containing the most commonly used functionality. Subsequent translation of proofs into Haskell is then specified rather trivially as our intermediate language for proof representation contains all required information, rendering the process of proof reconstruction superfluous.

A. Related work

Theorem provers are not a new thing. They have, however, come a long way ever since De Bruijn's Automath[4] in 1967. One of the problems that had to be solved (or at least ameliorated) was the way the proofs were conducted. Users had to construct proof terms by hand (basically sentences in a functional language reminiscent of formal lambda calculus). More recent provers have developed convenient notations used for construction of proofs, whether it is Coq-like imperative tactics languages[5], or Isabelle-like Mizar declarative approach[11]. Of course, there are also ways to mix and match the approaches. Microsoft's Lean[10] theorem prover is a good example thereof.

With regards to calculi used as a basis for subsequent analysis, there are generally the following options available. Either a dependently typed system is used – such as Coq's Calculus of Inductive Constructions[5](CIC) when considering "theorem proving – first" approaches, or Idris[3] dependently typed system from the perspective of environments that are first and foremost programming languages. Or, variations of Higher Order Logics(HOL) are used by tools such as Isabelle/HOL[11].

Concerning automation, there are many approaches available, whether they are based on heuristics or even being total resolution procedures for various subsets of calculi. First of, the closest procedure for inference to the approach we shall be utilizing is the one used in Djinn[1]. It is a freely available Haskell package capable of synthesizing Haskell implementations from types, based on intuitionistic propositional logic. However, the system cannot handle recursive data types, which on one hand, simplifies the proof search and removes the need for heuristics, but on the other, makes it extremely restricted. Even basic linear recursive structures such as lists are out of Djinn's reach.

Also, Coq comes with a variety of tactics that automate proving. We have been certainly inspired by the *auto* tactic, which works by introducing implication assumptions, ordering them by their cost and subsequently attempting to use them to solve the goal. This procedure however cannot deal with inductive case analysis. There is also the decision procedure *tauto*, which is described in Coq's documentation as "a decision procedure for intuitionistic propositional calculus based on the contraction-free sequent calculi LJT* of Roy Dyckhoff"[6]. The calculus description can be found here[7]. It can deal with a particular subset of product and sum types, namely the ones that correspond with logical connectives. For a more thorough inspection of Coq's automation facilities, one should refer to the Coq reference manual[6].

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

There is also a collection of resolution procedures and heuristics that Isabelle collectively calls Sledgehammer[2]. It works by invoking various automated theorem provers(ATP) on a requested goal. Considering ATPs, there is a plethora to choose from. They range from general purpose ATPs (such as [12]) to more focused ones[8].

Regarding code generation / extraction, most modern theorem provers offer facilities to extract executable implementations from constructive proofs in various languages. For example, Coq provides means of extraction to both Haskell and OCaml, using its functional Gallina proof terms as bases.

As for the overall prover design, it is generally the case that decisions are made in the initial concept phase of a prover. This removes some flexibility that a prover might otherwise have – while it is possible to change (or as it often happens, ”hack in”) the input language for proof construction, for instance utilizing Coq's tactic description language LTAC to specify tactics simulating declarative approaches, and also the output language, for example translating from one target language to an another. It is also common to allow user–defined procedures for resolving proof obligations, but the core principles (for example the unification algorithm) are locked in place.

B. Ingredients

The entire template can be conceptually divided into four parts, of which all are replaceable and / or adjustable based on the requirements we may have. Those may include (but are certainly not limited to) the specification and target language used or the nature and complexity of proofs we would like to conduct.

They are as follows:

- Calculus of deduction - a set of rules that forms the basis for analysis of propositions.
- Source language specification and mapping - definition of the input language along with a way of mapping its sentences into a form amenable to subsequent analysis; this generally amounts to constructing an implication / a function type.
- Analytical procedure - a method for automatically finding sequences of tactics ideally ending with axiomatic rules (that is, rules that hold vacuously), which corresponds with finding a constructive proof of a proposition in question. If one wishes to construct proofs manually, this step can be omitted.
- Generative procedure - a way to translate results of analysis into a target language.

This design slightly differs from what theorem provers normally use, as it explicitly modularizes the underlying theory used for analysis – this is achieved by a combination of allowing the process of analysis to be modified freely as well as enforcing utilization of an intermediate representation that reflects results of analyses.

C. Calculus of deduction

For the purposes of this paper, a System F based framework has been chosen. This decision stems from the fact that we are mostly interested in generating Haskell definitions which has also (rather unsurprisingly) influenced our choice of the target language. Haskell's type system is based on theory corresponding with our choice. In view of Curry Howard isomorphism, we are dealing with second-order logic limited to universal quantification over propositions, coinciding with type specifications containing parameterized types. The choice does not manifest within the ruleset directly and as such it is worth pointing out that rules requiring equality of types (such as the yet-to-be-introduced rule *trivial*) are capable of finding type unifiers (if they exist) using a first-order unification algorithm. As such, two types may be equivalent up to their common unifier, which is a set of substitutions that need to be performed for an exact match to occur. This idea of utilizing parameterization has been explored in a slightly different setting by Philip Wadler[14].

A word on conventions and notations. Rules are of form *name<parameters...>(inference)*, where a rule name corresponds with an operation performed on a signature and parameters contain additional information required for the subsequent synthesis. The idea of tracking the analysis using a meta-language that chains fully descriptive atomic steps thereof allows us to split the design of the prover into modules. As a consequence of this choice, we can construct a proof using any source language within any framework of deduction as long as we specify its translation into the intermediate proof language. Naturally, we can subsequently also transform proofs into a target language of our choice. ”Inference” is used to denote each separate branch of analysis should more than one be required (for instance when conducting separate analysis for every single tag of a discriminated union / sum type). A hypothesis h_i of type A is denoted $h_i : A$. Context remainder marked Γ is always assumed to be disjoint with a potentially empty set of explicitly listed hypotheses and is notationally separated from conclusion by $\vdash$. We use the term ”tactic” interchangeably with ”rule”. Lastly, we generally prefer terminology from the area of type theory rather than its ”pure” logic twin.

The set of deduction rules:

1) Trivial: The *trivial* tactic represents a relaxed form of reflexivity without the possibility of contraction of hypotheses in context. In other words, it allows us to use a function argument in the definition of a function, as long as it can be uniquely identified. It only has a single parameter, namely a hypothesis. Do note however, that it is defined only for contexts with exactly one hypothesis that matches the type of conclusion.

$$[\nexists k \in \mathbb{N} : (k \neq i \wedge (h_k \in \Gamma : A))] \frac{}{h_i : A, \Gamma \vdash A} \text{trivial}{<}h_i{>}$$

2) Intro: The *intro* tactic introduces implication. That is, we assume the first argument has been supplied and can therefore be moved into the context, which represents our available hypotheses. The tactic is unary with the parameter representing the name of the introduced hypothesis.

$$[h_i : \_ \notin \Gamma]\,\frac{h_i : A, \Gamma \vdash B}{\Gamma \vdash A \to B}\,\text{intro}{<}h_i{>}$$

3) Clear: The *clear* tactic represents context weakening. It removes a hypothesis, which is the sole parameter of the tactic.

$$\frac{\Gamma \vdash A}{h_i : \_, \Gamma \vdash A}\,\text{clear}{<}h_i{>}$$

4) ElimSum: ElimSum is a form of disjunction elimination. It considers all constructors of an arbitrary sum type, effectively splitting the process of analysis into multiple isolated branches, one for each of the data constructors of the type in question. It has a parameter denoting the eliminated hypothesis and it also makes note of the data constructor chosen for each of the branches.

Let the number of c constructor arguments be $N\ c$. And let $M\ c\ r$ be $a_0^c \to ... \to a_{Nc-1}^c \to r$ for $Nc > 0$, otherwise r.

$$\frac{h_i : A, \Gamma \vdash M\ c_0\ B \qquad ... \qquad h_i : A, \Gamma \vdash M\ c_{n-1}\ B}{h_i : A, \Gamma \vdash B}\,*$$

Where n is the number of constructors of type A and * is *elimSum<h_i>($c_0^A ... c_{n-1}^A$)*.

5) Apply: The *apply* tactic is a variation of implication elimination. It effectively calls a function and creates a new branch of analysis for every one of its arguments. The goals of the newly introduced branches correspond with types of the arguments of the applied function. The tactic has a parameter denoting which hypothesis has been used and it also contains information about which argument of the hypothesis in question the new inference branch belongs to.

Let $\rho\, f$ be the arity of *f*.

$$\frac{h_i : \alpha^A, \Gamma \vdash \beta \qquad ... \qquad h_i : \alpha^A, \Gamma \vdash \omega}{h_i : \alpha^A, \Gamma \vdash A}\,\text{apply}{<}h_i{>}(0...ar/ar)$$

Where ar is $\rho\ h_i - 1$ and α^A is either a function or a data constructor of any arity ultimately returning *A*, that is, either takes the shape of *A* (in which case the rule is an axiom) or $\beta \to \gamma \to ... \to \omega \to A$.

D. Source language specification and mapping

We provide EBNF for a specification language used for the purposes of the paper. It is concise and is tailored to work closely with the chosen calculus of deduction. This means that every syntactic construct can be taken advantage of (in terms of its semantic content) and is also capable enough to represent everything the calculus can handle. More specifically, we are focused on obtaining information from the following algebraic datatypes:

- Sum types - See the *elimSum* rule as their eliminator, and *apply* as their constructor.
- Product types - See the *elimSum* rule as their eliminator (with the corresponding expansion of *M*), and *apply* as their constructor.
- Function types - The pair of *intro* and *apply*.

Terminals that could be confused with EBNF notation are enclosed in apostrophes.

```
IDENT     ::= < identifier >
TPARAM    ::= IDENT | ‘(‘ FUNCTYPE ‘)‘ | ‘(‘ DATATYPE ‘)‘
TPARAMS   ::= TPARAM +
DATATYPE  ::= IDENT [ TPARAMS ]
FUNCTYPE  ::= TYPE –> TYPE ( –> TYPE )∗
TYPE      ::= DATATYPE | ‘(‘ FUNCTYPE ‘)‘
KIND      ::= IDENT
PARAMS    ::= parameterization { [ KIND ( , KIND )∗] }
OPTYPE    ::= : FUNCTYPE
OP        ::= IDENT [ OPTYPE ]
OPS       ::= operations { OP ( , OP)∗ }
SPEC      ::= Specification IDENT { [ PARAMS ] OPS }
```

Listing 1. Specification - Data types

```
IDENT      ::= < identifier >
DATATYPE   ::= IDENT [ TPARAMS ]
FUNCTYPE   ::= TYPE ( –> TYPE )∗
TYPE       ::= DATATYPE | ‘(‘ FUNCTYPE ‘)‘
SIGNATURE  ::= IDENT : FUNCTYPE
OPERATION  ::= SIGNATURE
```

Listing 2. Specification - Functions

Given the language has been created for this exact purpose, the mapping is straightforward. Data types (*Listing 1.*) contain explicit parameterization and operations, while propositions can only be (potentially nullary) function types even on the level of syntax (see *Listing 2.*). This happens to play nicely with the operations we have available. Any formal or semi-formal definition of semantics of the language would be somewhat redundant as the constructs only serve to hold type information and as such simply describe families of algebraic data types and type signatures of functions. Of course, if we wanted to use an existing specification language (for example a mainstream programming language), the process would be more involved assuming the underlying calculus would remain unchanged.

E. Analytical heuristic

We have decided to take the approach of full automation of proof search utilizing heuristics (as opposed to strict resolution procedures), allowing us to, for all intent and purposes, sometimes synthesize a definition for a signature under ambiguous circumstances. Considering our focus on Haskell functions, the proving strategy is tailored as follows:

We begin with assuming (adding to context of available hypotheses) the initial form of the signature to be analysed; this effectively allows us to invoke the function recursively with the $apply$ tactic should we so desire. We also assume all data constructors of user defined types at our disposal.

The first tactic tried is $intro$, immediately followed by $trivial$ in an attempt to complete the analysis. If invocation of $trivial$ succeeds, we $clear$ the hypothesis and halt.

Otherwise we attempt to deconstruct the most recently introduced hypothesis using $elimSum$, after which we $clear$ the eliminated hypothesis and restart the entire procedure on every new branch of the analysis, should any be introduced by the $elimSum$ tactic. The immediate clear on use captures the idea that we do not want to reuse function arguments. It is worth noting that extra care needs to be taken for user defined recursive data types. For example, given a non-empty list data constructor $Cons : a \to List\ a \to List\ a$, eliminating a hypothesis into this constructor could form an infinite cycle of deconstruction of the second argument. This problem can be to some extent solved by bounding the maximum depth of allowed eliminations.

If no progress can be made, we continue by attempting to invoke $apply$ on one of the hypothesis available which is of function type and whose return type (of its uncurried form) is equal[1] to the type of conclusion of the relevant inference branch. If $apply$ succeeds, the $clear$ tactic is called to remove the applied hypothesis. In the case where there are multiple viable hypotheses to $apply$, we assign a cost to each of them based on their arity; the lower it is, the higher the cost and thus they are less likely to be selected. It is important to note that data constructors for types are considered functions as well. If we cannot uniquely identify a function to apply with this criteria, we choose one arbitrarily from the ones that cost the least. Lastly, the recursive invocation of the original function has always the highest cost regardless of its arity. Whenever a new branch of analysis is created, the sequence of previously generated atoms is prefixed to it.

```
Type = Scalar | Parameterized | Function
Context = [(String, Type)]
Env = (Type, Context)

inferImplAux : (seen : [Env]) -> (env : Env) -> [Env]
inferImplAux seen env =
  let continue e = case applyAux e of Nothing -> []
                                      Just result -> result
      (i, env')  = intro env
      (t, env'') = trivial env'
        envs     = map (clear i) $ elimSum env'' i in
          if env `elem` seen then continue env
          else case env'' of Nothing ->
                               concatMap (
                                 inferImplAux (env :: seen)
                               ) envs
                             Just (h, _) ->
                               continue $ clear env'' h

applyAux : (env : Env) -> Maybe [Env]
applyAux env =
  let len = length . snd $ env
      s   = sortBy cost desc in
   if len < 1 then Nothing
   else let envs = apply env
                     (snd . last . s . snd $ env) in {
     case envs of Nothing -> applyAux shorterEnv
       where shorterEnv = take (pred len) . snd $ env;
         Just (h, envs') =
           let envs''   = map (clear h) envs'
               envs'''  = map trivial envs''
               solved   = filter isJust envs'''
               unsolved = filter isNothing envs''' in
                if null unsolved
                then Just $ concatMap fromJust solved
                else let result =
                  concatMap inferImplAux [] unsolved
                  in if not . null $ result then Just result
                     else Nothing
   }

doAnalysis = inferImplAux []
```

Listing 3. Pseudocode for the proof strategy, starting with doAnalysis

F. Generative procedure

As already stated, our target language of choice is Haskell. Proof script translation is fairly easy to conduct as the rules representing steps of analysis explicitly carry all the required information and can be processed in isolation.

Relationships between analytical rules and their representations in Haskell:

Tactic	Analytical atom	Generative atom
trivial	trivial<h>	h
intro	intro<h>	\h →
clear	clear<h>	
elimSum	elimSum<h>(c0 ... cn)	case h of c0 ... cn →
apply	apply<h>(0 ... n)	h 0 1 ... n

TABLE I. ANALYTICAL ATOMS AND THEIR GENERATIVE COUNTERPARTS

All we need now is a way to introduce an equation with name h. For Haskell it is as simple as "$h =$". As we will see in an example later, this simplistic approach will yield certain artefacts that will need mending.

II. EXAMPLE - CURRY

Let us take a look at the sum of the aforedescribed parts, using the curry function as an example. Before we move on however, let us first define the notion of *pair*, the most basic of product types, using our specification language.

```
Specification Pair {
    parameterization {
        Any a,
        Any b
    }
    operations {
        Make : a -> b
    }
}
```

Do note that the operation Make (the constructor of Pair) does not need to mention its return type (as it can be inferred). As such, it is a binary operation returning a *Pair* parameterized by two types, *a* and *b*.

We are now ready to declare the curry function itself:

```
curry : ( Pair a b -> c ) -> a -> b -> c
```

Following the proof construction strategy, we construct an initial environment consisting of the *curry* function itself (for the purposes of recursive invocation) and of constructors for all the user defined data types in the type signature - in this case, the *Make* constructor of *Pair*. With hypotheses out of the way, we now set the conclusion to be the type of the function in question: $(Pair\ a\ b \to c) \to a \to b \to c$

Compressing the initial hypotheses into Γ, we get:

$$\frac{}{\Gamma \vdash (Pair\ a\ b \to c) \to a \to b \to c}$$

[1] Equal for whatever definition of equality we have at our disposal

Now the proof construction continues as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{y:b,\Gamma'\vdash}{y:b,x:a,\Gamma'\vdash}\text{clear<x>}}{y:b,x:a,\Gamma'\vdash a}\text{trivial<x>}}{y:b,x:a,\Gamma\vdash a}\text{clear<Make>}\qquad\dfrac{\dfrac{\dfrac{x:a,\Gamma'\vdash}{y:b,x:a,\Gamma'\vdash}\text{clear<y>}}{y:b,x:a,\Gamma'\vdash b}\text{trivial<y>}}{y:b,x:a,\Gamma\vdash b}\text{clear<Make>}}{y:b,x:a,\Gamma\vdash \mathit{Pair}\ a\ b}*}{y:b,x:a,f:\mathit{Pair}\ a\ b\rightarrow c,\Gamma\vdash \mathit{Pair}\ a\ b}\text{clear<f>}}{y:b,x:a,f:\mathit{Pair}\ a\ b\rightarrow c,\Gamma\vdash c}\text{apply<f>(1/1)}}{x:a,f:\mathit{Pair}\ a\ b\rightarrow c,\Gamma\vdash b\rightarrow c}\text{intro<y>}}{f:\mathit{Pair}\ a\ b\rightarrow c,\Gamma\vdash a\rightarrow b\rightarrow c}\text{intro<x>}}{\Gamma\vdash(\mathit{Pair}\ a\ b\rightarrow c)\rightarrow a\rightarrow b\rightarrow c}\text{intro<f>}$$

Where * is *apply<Make>(1/2) / apply<Make>(2/2)*.

A few important remarks on the analysis. When *apply<Make>(x / 2)* is invoked, the entire tree is actually split into two identical new ones (up to the invocation of the rule that had initiated the split), with the left tree having *apply<Make>(1 / 2)* and the right tree having the remaining one. This might seem somewhat redundant but it is a necessary consequence of the fact that the entire stream of analytical atoms can be handled in isolation and as such the information of the order of function arguments order must be preserved. For brevity and space constraints, we decided to keep the analysis in a single tree splitting into two subtrees. It represents the idea in a more concise way, even if it is in some sense incorrect with regards to the actual analysis that takes place. Note that * is therefore different in the hypothetical two trees.

An another important thing to note is that *clear<x> / clear<y>* in left and right tree respectively has zero impact on the process of analysis and an argument could be made that its omission would only bring benefits. However, referring to the context oblivious nature of the analysis, we have no way of telling whether a solution found with *trivial* is a solution to the original proof goal or one of its subgoals created by *apply* or *elimSum*. As such, *clear* is simply always included once a hypothesis has been used.

Also, the step of *clear<Make>* changing Γ to Γ' might appear magical. However, remember the construction of the initial environment Γ. We included *Make* as it is one of the constructors of a user defined data type we are working with. As such, we could state that $\Gamma' = \Gamma\backslash(Make : a \rightarrow b \rightarrow Pair\ a\ b)$.

Now that the analysis has been finished and all goals have been solved, we end up with the following set of strings of atoms in the left tree:

```
intro<f> intro<x> intro<y> apply<f>(1/1)
clear<f> apply<Make>(1/2) clear<Make>
trivial<x> clear<x>
```

And in the right tree:

```
intro<f> intro<x> intro<y> apply<f>(1/1)
clear<f> apply<Make>(2/2) clear<Make>
trivial<y> clear<y>
```

Do note that no tree actually ever branches because in such cases, the analysis simply creates a new one. As such, linearisation at this stage is not necessary (in some sense, it has already happened as the analysis was being conducted). We are now ready to transcribe the atoms of inference into the target language.

The translation of the left and right tree respectively:

$$curry = \backslash f \rightarrow \backslash x \rightarrow \backslash y \rightarrow f\ Make\ x \tag{1}$$
$$curry = \backslash f \rightarrow \backslash x \rightarrow \backslash y \rightarrow f\ Make\ y \tag{2}$$

There are four issues arising using this simple approach, three of which surface looking at the generated (and incorrect) Haskell implementation of *curry*:

1) We have multiple definitions of the same function.
2) Function calls are not properly parenthesized as atoms are translated in isolation.
3) Functions with more than a single argument in their uncurried form are mishandled.
4) In case of translating *elimSum*, we emit $case\ h\ of\ Constructor \rightarrow \backslash arg0 \rightarrow \backslash arg1 \rightarrow ...$; however, Haskell does not allow us to use partially applied constructors and as such, we need to contract the above transcription to $case\ h\ of\ Constructor\ arg0\ arg1... \rightarrow ....$

Given the fact that all function applications carry their argument order with them and the fact that all sum deconstructions make note of the constructor they are a branch of, all of the problems can be trivially fixed within the implementation and deserve no further description, partly due to space constraints.

We can now finish our Haskell implementation of *curry* by merging the call to $Make$, inserting parentheses around every function call (both f and $Make$) and removing their common prefixes. Do note that some parentheses shall be redundant if function application is at the top level.

```
curry = \f -> \x -> \y -> (f (Make x y))
```

This is indeed a correct implementation of Haskell's $curry$ function from Prelude[2].

III. More of the Prelude

We shall list a few functions that the prover has been tried with and provide insights on what had gone wrong in case the resulting implementation is not exactly what we are interested in. That said, results always typecheck (if there exists at least one, even if incorrect solution) and as such can be used as a template to finish the implementation manually.

```
const : a -> b -> a
const = \x -> \y -> x

compose : (b -> c) -> (a -> b) -> a -> c
compose = \f -> \g -> \x -> f (g x)

flip : (a -> b -> c) -> b -> a -> c
flip = \f -> \y -> \x -> f x y
```

[2] Haskell has built-in notation for its standard tuples, which is in our case replaced with an explicit *Pair* data type

```
maybe : b -> (a -> b) -> Maybe a -> b
maybe = \x -> \f -> \mx -> x
```

Assuming we have the standard definition of *Maybe* (see Prelude[9]), we still do not quite get what we want. The heuristic is very eager to prove propositions and as such, when it reaches the state where *b* is needed, it simply uses the default value right away. No case analysis is done on the hypothesis *mx*. Indeed, the calculus we have decided to choose is not expressive enough to state in its type that the two resulting cases should be different, as the only universal quantification can be done over types. To even have the ability to encode such invariant, we would need access to dependent types.

```
snd : Pair a b -> b
snd = \x -> case x of Make xx xy -> xy

uncurry : (a -> (b -> c)) -> Pair a b -> c
uncurry = \f -> \x ->
    case x of Make xx xy -> f xx xy

map : (a -> b) -> List a -> List b
map = \f -> \xs -> case xs of
                       Nil -> Nil
                       Cons x ys -> Nil
```

Assuming we have the standard definition *List* (see Prelude[9]), the result is still not exactly what we want. The problem is very similar to the one we have already encountered with *maybe*; the heuristic is very eager. We see that we need *List b* and we can very easily construct one with the low cost nullary constructor *Nil*. And we do just that. Again, there is not much that can be done about this if we do not have access to a way to express additional invariants - for example in this case, an additional constraint on equality of lengths of input and result list would be sufficient for this very heuristic to find the proper implementation of the standard map. An attentive reader might notice that in the *maybe* case, we did not end up deconstructing the sum typed argument, unlike in here where the list has been analysed by constructors. This is because the former required *trivial* and the latter *apply* which just happened to have no arguments; all in accordance with the heuristic.

IV. Discussion

We have designed a fully modular theorem prover that allows us to combine various input methods with arbitrary output methods utilizing any process of analysis within any underlying framework. The key idea that allows for this approach is to employ a separate intermediate representation for results of analyses. While we are not aware of any other prover using an atom meta–language for proof result descriptions, it is worth noting that there do exist common formats for definition and proof storing, sharing and reconstruction (a very good overview is available at [13]) as well as the already described Coq–like tactic proof scripts – these differ slightly from our approach in the fact that the entire proof state is completely implicit, as they are usually aimed at interactive theorem proving with explicit proof state feedback from a prover, separate from the proof script itself.

We have then created a particular combination of various modules with the intent to automatically synthesize implementations of Haskell declarations; this has, naturally, influenced the choices we have made for each of the parts of the prover. First of, in this particular case, we have managed to completely avoid the discussion about how to interact with the prover when it comes to constructing proofs, as the entire process is completely automatic. Of course, it is still necessary to have a way to specify function type signatures and data type declarations – for this purpose, we have designed a simple input language.

Secondly, we have chosen a System F–based deductive calculus, which is slightly out of the ordinary, as its expressive power is nowhere near acceptable levels with regards to general purpose theorem proving. As we have demonstrated by creating a proof of concept implementation followed by an attempted generation of Haskell's standard Prelude[9], even when only type parameterization is present, that is, a rather simplistic underlying theory is used, there is enough information in signatures to either get an implementation that does exactly what we want, or at least obtain a template that can be manually completed to attain the intended functionality. While System F appears to be reasonably sufficient for this particular use of the prover, judging by the number of signatures successfully converted into correct implementations, the entire process would most certainly benefit from introduction of a variation of dependent type theory, as we have touched on in the discussion on *map* in section 2. The reason for choosing System F is twofold. First, we wanted to stay "true" to Haskell's type system – tangentially leading to an interesting discussion about the need of a more expressive type system to describe a less expressive one and just how far can a simpler approach be taken. The second reason is a lot more practical – ease of implementation; perhaps pilfering parts of more mature theorem provers and stitching them together in our framework would make for an interesting study.

With regards to proof construction, we have chosen a heuristic analytical procedure, which, in combination with the chosen deductive calculus is capable of synthesizing a reasonable subset of Haskell's Prelude[9], giving us confidence in its practical applicability, as it had not been crafted with a particular set of hypotheses in mind, but rather has been tested with functionality Haskell programmers use on daily bases. It is also worth reiterating that if we kept the procedure intact and simply enhanced the calculus (and the unification algorithm) with dependent types, the subset of solvable propositions would be much greater, as it would be possible to specify invariants more precisely within type signatures of functions. In general, the heuristic procedure is flexible enough to take advantage of more expressive calculi – this is mostly because it treats notion of equality transparently, allowing it to discriminate between candidate terms with higher precision if the candidates themselves carry more information. An "apology" should be issued with regards to vagueness of what subset of propositions can be handled. Given our heuristic approach that uses function arity for costs and chooses arbitrarily in case of ambiguity, it is difficult to describe and quantify the set of solvable propositions. This question is certainly subject to further inspection.

Concerning code generation / extraction, we have chosen a functional output language, simplifying the process of translation even further, allowing us to "get away with" handling each atom in isolation using a simple translation table. This has also been made possible by the fact that analytical atoms themselves have been chosen to be entirely self contained and enriched with all information the process of analysis had inferred.

Acknowledgment

The paper was supported by project KEGA no. 079TUKE-4/2017.

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-15-0055.

We would also like to thank anonymous referees for their suggestions.

References

[1] L. Augustsson, *Djinn*. [online]. https://hackage.haskell.org/package/djinn

[2] J. C. Blanchette, *Hammering Away A Users Guide to Sledgehammer for Isabelle/HOL*. [online] https://isabelle.in.tum.de/dist/doc/sledgehammer.pdf

[3] E. Brady, *Idris, a general-purpose dependently typed programming language: Design and implementation*, Journal of Functional Programming, 23, 552–593, 2013

[4] N.G. de Bruijn, *Description of the language AUTOMATH*, Department of Mathematics, Eindhoven University of Technology, 1967. [online]. https://www.win.tue.nl/automath/, document XAUT001

[5] The Coq development team, *The Coq proof assistant*, 2004. [online]. http://coq.inria.fr

[6] The Coq development team, *The Coq proof assistant reference manual*, 2004. [online]. https://coq.inria.fr/distrib/current/refman/

[7] R. Dyckhoff, *Contraction-free sequent calculi for intuitionistic logic*. The Journal of Symbolic Logic, 57(3), 1992.

[8] M. Ganesalingam and W. T. Gowers, *A Fully Automatic Theorem Prover with Human-Style Output*, Journal of Automated Reasoning, Berlin, Germany: Springer-Verlag, 2017

[9] S.P. Jones, *The Haskell 98 Report* [online]. https://www.haskell.org/onlinereport/standard-prelude.html

[10] L. d. Moura and S. Kong and J. Avigad and F. v. Doorn and J. v. Raumer, *The Lean Theorem Prover (system description)*, 25th International Conference on Automated Deduction (CADE-25), Berlin, Germany, 2015. [online]. https://leanprover.github.io/papers/system.pdf

[11] T. Nipkow and M. Wenzel and L.C. Paulson, *Isabelle/HOL: A Proof Assistant for Higher-order Logic*, Berlin, Germany: Springer-Verlag, 2002.

[12] S. Schulz, *E - A Brainiac Theorem Prover*, Journal of AI Communications 15 (2/3), 111126. 2002

[13] G. Sutcliffe, J. Zimmer, S. Schulz, *TSTP Data-Exchange Formats for Automated Theorem Proving Tools*, 2004. [online]. http://www4.in.tum.de/~schulz/PAPERS/SZS-DMA-2004.pdf

[14] P. Wadler, *Theorems for free!*, 1989. [online]. http://ttic.uchicago.edu/~dreyer/course/papers/wadler.pdf

Survey of traffic prediction methods for dynamic routing in overlay networks

Jindrich Skupa, Jiri Safarik
Department of Computer Science and Engineering
University of West Bohemia
Pilsen, Czech Republic
Email: skupaj@kiv.zcu.cz

***Abstract*—We introduce and compare methods for traffic prediction focused on dynamic routing in overlay networks and their practical usage. Overlay networks often need special routing, even when underlay networks do routing on their own. There are more unknown variables then in traditional routing approach. Dynamic routing is based on several metrics and prediction could be judged by many statistical values. All introduced methods are compared by computing costs, implementation complexity and statistical values. Main statistical quantity is MAE (mean absolute error), which presents accuracy of prediction. We apply prediction on routing model to increase quality of service. So, we can focus on final route parameters not only on prediction accuracy. Naturally, main route parameter is transfer speed, but in real life scenarios we can find examples, where it is not only one metric. Let us assume real time audio/video stream, where we are interested in lowest range of scatter to reduce flapping of stream quality. In the paper, we will discuss impact of used metrics and algorithms on final quality of service.**

1. Introduction

Routing and forwarding are fundamental tasks for Internet function. Internet service providers (ISP) have to transport the huge amount of data because internet content grows up rapidly. There are many and many new services which offers multimedia content and online streaming services. This paper will be focused on online streaming services like a phone call, teleconference and live streaming. Internet service and content providers build high-speed transport lines and create sophisticated environments to delivery data with quality of service assurance. Unreliability can cause huge income losses and negative impact on reputation. Modern content providers have following requirements for transport system: reliability, high transport speed, high throughput, low latency and low jitter.

There is no universal solution how to create the reliable infrastructure which is able to reach this requirement. Usual architecture is the creation of content delivery network(CDN). The content delivery network contains servers distributed geographically near to end users. These servers form service overlay network. Data are routed using traditional routing protocols like BGP for external or OSPF for internal traffic. In many cases, there are used custom routing algorithms suitable for overlay networks focused on important metrics. Used metric or combination of metrics depends on the carried traffic class.

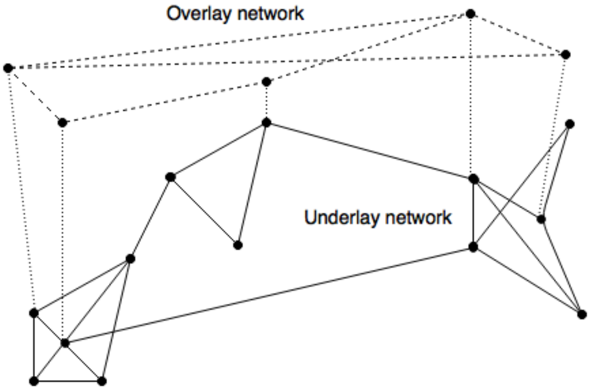


Figure 1. Overlay network

Each traffic class has different needs. When multimedia files are delivered, delivery time is most important, which implies high transfer speed. In other cases like a live stream or conference call, jitter and delay need to be minimized.

2. Overlay Networks

Overlay networks are built over underlay networks. Underlay network is usually the Internet based on IP protocol. Each node of overlay network uses its own internet connection to connect to each other node. In case, there are some restrictions on a path between nodes, the overlay network is used as a bridge over another node.

Overlay networks are build in many ways for many purposes. Overlay network could act as data storage (for example DHT peer to peer networks), as data routing infrastructure (old Skype protocol), as high availability environment, a single computation environment and many other. Example of an overlay network is in Figure 1.

Overlay networks may have many topology organizations, they could be organized around central node which acts as the master. Another topology could be masterless, peer to peer, where each node has connections to each other.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

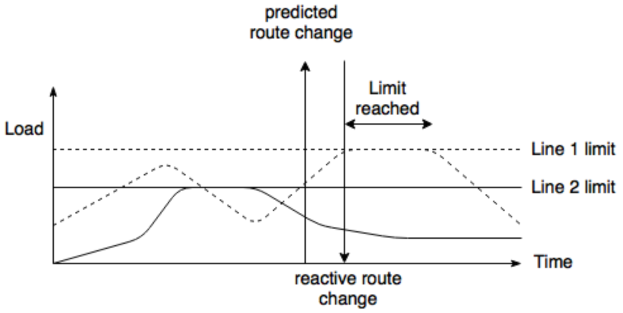


Figure 2. Reactive vs proactive routing

Next topology could be a ring, where each node knows only its neighbours. The overlay network topology is fully independent on underlay network. The topology is chosen by network purpose and is more suitable then underlay network [1] [2].

In this paper, we assume the peer to peer topology where nearly each node has a direct connection to each other. This topology fits the best to our research area. We assume content delivery network for content streaming with multiple data source or teleconference stream without the central point. Replication and signalling processes are out of the scope of this paper.

3. Prediction Motivation

This paper focuses on network traffic prediction based on several different prediction approaches. The main motivation for network traffic prediction assumes that it is better to change the route before it is too late. We optimize routing table changes to obtain the better statistical quality of traffic than by classical reactive routing. The assumed result is cut to off extreme values of traffic routing. Let us assume two independent lines in Figure 2. Reactive routing reacts on the saturation of line and changes the route when the limit is reached, but pro-active routing based on the network traffic prediction changes route in advance to avoid the limit is reached.

The current route throughput could be lower for a short term. In the previous paragraphs, we decided that we are focused on stream stability, not the absolute network throughput or transfer speed. Modern streaming services are able to change stream quality, but for viewers, it's uncomfortable to do it very often. Our approach is to decrease the number of stream quality switching to get the better experience with stable stream transport. Stream quality switching usually causes the stream tearing with a small time of blackouts.

Our approach is focused on network quality jitter causing stream quality flapping and bad user experience.

4. Prediction Methods

The network load and throughput measured in constant time interval create time series. The time series could be analyzed to obtain its statistical qualities. Based on time series properties and behaviour in short and long-term history we can predict the near future value with some level of probability [3] [4].

In computer science are several prediction algorithms for time series which are widely used. The algorithms could be compared by several metrics. Run-time or memory complexity is one of them. The other point of view is algorithm accuracy or mean square or absolute prediction error [5]. We are using network traffic prediction to find out optimal routing algorithm to get better statistics qualities.

Following the main algorithm, classes are chosen to compare and to validation of our premise. Proposed methods are widely used for time series prediction. Network traffic prediction is similar to other time series. The methods used in different research area depend on traffic or time series characteristics.

4.1. ARMA Methods

The first is mathematical and statistical based method called ARMA [6]. This method is based on properties of short time series history. ARMA refers to an autoregressive moving average method, which is a combination of autoregression and moving average. Autoregression (AR) expresses the next time series value depends on the linear combination of values in near past. Moving average (MA) says the autoregressive error depends on the linear combination of previous errors. Formal math formula is on Equation 3.

Each part is parametrized, $AR(p)$, parameter p is a number of previous values, vector y of previous values the size of p is multiplied by the vector of the same size ϕ. Vector ϕ represents the linear combination of previous values. The formula is described on Equation 1.

Second part $MA(q)$ has parameter q which stands for the size of vectors e of previous errors and Θ, the multiplication of these vectors takes error moving average. Vector Θ represents the linear combination of previous errors.The formula on Equation 2 describes whole relation.

The combination both parts is expressed by the formula on Equation 3. The future value is based on the sum of the linear combination of previous values and the linear combination of previous errors.

The c stands for constant and e_t stands for random value with Gauss distribution and zero mean value. ARMA method needs to estimate of $ARMA(p, q)$ parameters and the values of ϕ and Θ vectors based on known time series values. Values of ϕ and Θ vectors should be continuously calculated to adopt new values of time series.

ARMA method is popular in statistics and econometrics and there are plenty extensions and variants, like $ARIMA$ (Autoregressive Integrated Moving Average) [7] or $NARMA$ (Nonlinear Autoregressive Moving Average) for non-linear systems or $ARFIMA$ (Autoregressive Fractionally Integrated Moving Average) or $ARMAX$ (Autoregressive Moving Average Model with Exogenous Inputs Model).

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + e_t \quad (1)$$

$$y_t = c + e_t + \Theta_1 e_{t-1} + \Theta_1 e_{t-2} + ... + \Theta_q e_{t-q} \quad (2)$$

$$y_t = c + e_t + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=0}^{q} \Theta_j e_{t-j} \quad (3)$$

$ARMA$ method or variants were chosen because of quite a simple implementation and computing complexity. When we use $ARMA$ based methods the main problem is fitting the model, the way of choosing parameters p and q or other parameters regarding chosen method.

4.2. Neural Networks

Another very popular methods for time series predictions are neural networks [8] [9]. The neural network consists of neurons connected into oriented and weighed graph which is divided into layers. The basic neural network has one input layer, one output layer and a couple of hidden layers. The size of input or output layer depends on a size of input or output. The base unit is neuron.

Arrangement, size and count of a hidden layer depend on kind of neural network. Edges are divided into two groups based on their direction. In the direction from input to output, they are forward or backward in opposite way.

There are several kinds of neural network user for prediction. Current research and applications prefer LSTM (Long Short-Term Memory) networks, which are the special case of recurrent neural networks (RNN) with an ability to learn long-term dependencies [10].

The advatage of LSTM is ability to identify nonlinearity and randomness in traffic load. This advatage offers higher accuracy of predicted values. The LSTM is also less dependent on back-propagated error caused by memory. Lower back-propagated error dependency fits better to use long term traffic prediction and observation.

Other variants of neural networks (Elman Neural Network, Multi-layer Perceptron) are popular in current research. Chosen neural network depends on use cases regarding defined needs, like energy efficiency, computation costs or implementation demands.

4.3. Other Methods

Current research is mainly focused on methods mentioned in previous paragraphs, but there are much more methods. Our research is focused on the most popular methods which have the higher probability of success. There is a list of methods also used for traffic prediction, Hidden Markov Models, Support Vector Machine, Wavelet Analysis, Genetic Algorithms, Chaos Theory Based Prediction and Fuzzy Logic Based Algorithms [11] [12] [13] [14].

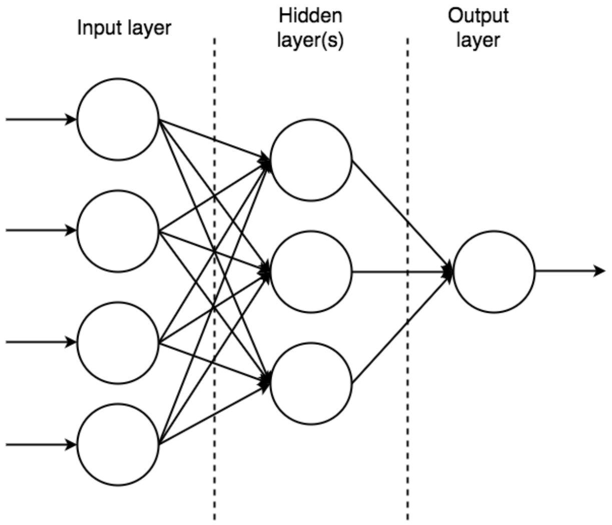


Figure 3. Multi-layer Neural Network

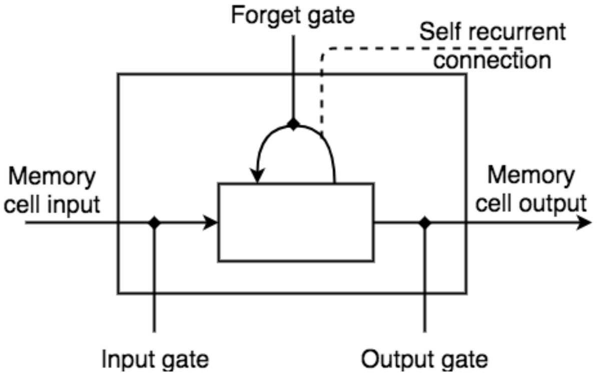


Figure 4. LSTM Memory Cell

There are many hybrid methods based on the combination of previous methods, where some of them are used in the chain with other ones and some of them are used for heuristics or correction of the main algorithm [15].

5. Experiments

To compare chosen methods we prepared following experiment. Let us assume one data stream source and one data stream destination. Between these nodes is overlay network consisting of four nodes connected one to each other. Source and destination node have two entry connections into overlay network. Schema of experiment overlay network is in Figure 5.

All lines between nodes are independent. We will use prediction algorithms to predict future load on each of them and apply routing algorithm to get the shortest path from source to destination. Shortest path means the path with the lower load.

In each step, we run prediction algorithm and routing algorithm to get the shortest path. When we found the

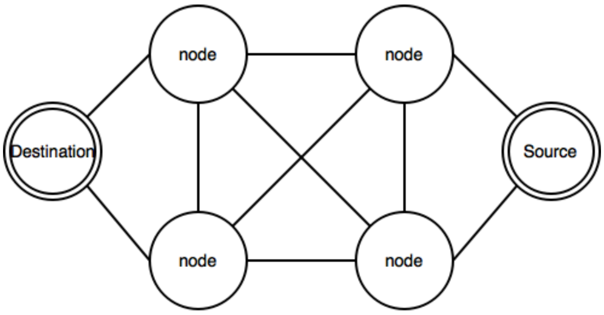


Figure 5. Experiment overlay network

```
for line in connections do
  for alg in algorithms do
    predict_line_load(alg, line)
for alg in algorithms do
  get_routes(alg, predicted, dest)
  use_route(alg, predicted, dest)
calculate_metrics(routes)
```

Figure 6. Experiment pseudocode

shortest path, the routing table is reconfigured and the new path is used. The path between source and destination is loaded with the constant load to simulate constant bit rate of audio and video stream. The load is the single direction. This load is added to used lines load. Experiment is described in pseudocode in Figure 7.

The experiment is realised as computer-driven simulation written in Python using libraries implementing chosen statistical methods and neural networks.

Input data for each line were obtained on routers in campus network to get real-life data for computations. The rate of samples is 1s and for each line, we had 1000 samples. The experiment was realised in the 150s window on the end of time series.

Neural network based prediction using LSTM and MLP and ARIMA methods were chosen to compare prediction methods and prove proposed approach in proactive dynamic routing in overlay networks. These methods were chosen because are used in the majority of current research.

6. Results

The results in Table 1 show statistical qualities of prediction and summary routed traffic properties. Relative values compared to original routing are in Table 2. Shown qualities are the mean and median load of used lines, standard deviation and variance and route change count during 150 s window used in the experiment.

Prediction methods on selected line are shown in Figures 7,8,9. Neural network based prediction methods ares in Figure 7. ARIMA prediction method result is in Figure 8. Both figures are showing prediction on the same line on same interval.

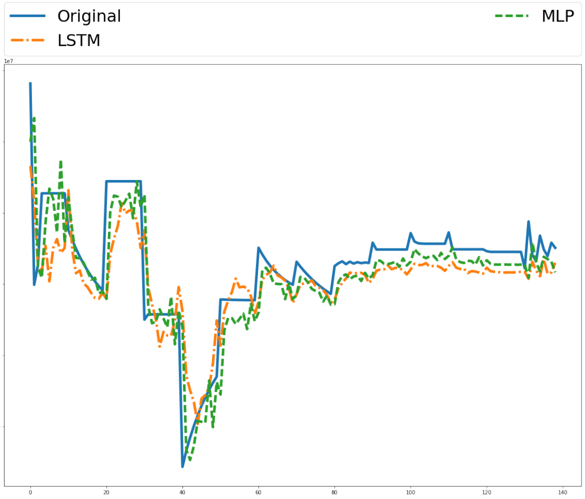


Figure 7. Prediction with LSTM and MLP NN

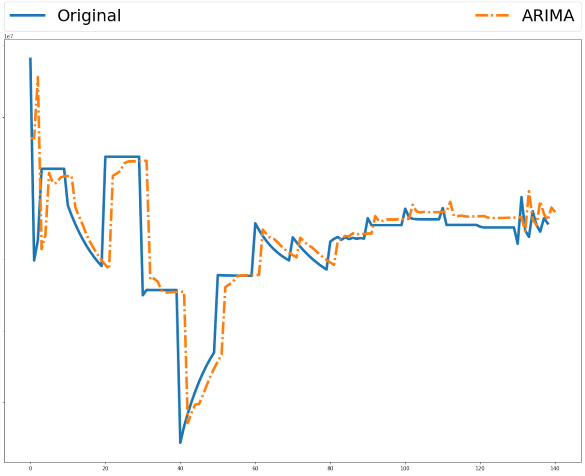


Figure 8. Prediction with ARIMA

Final routes based on the prediction of each line load are shown in Figure 9. Final route was estimated based on predicted load values, the figure is showing final route load which consists of original values.

In Table 2 is visible that proposed approach filled our expectations only for ARIMA method. Other prediction methods fail. ARIMA methods filled our expectations regarding standard deviation (STD) and variance (VAR), but median and mean load is 1.4 higher than when reactive routing is used. This method is using more stable lines but with higher load.

Neural network based prediction method bring higher standard deviation and variance using lines with the higher load in total.

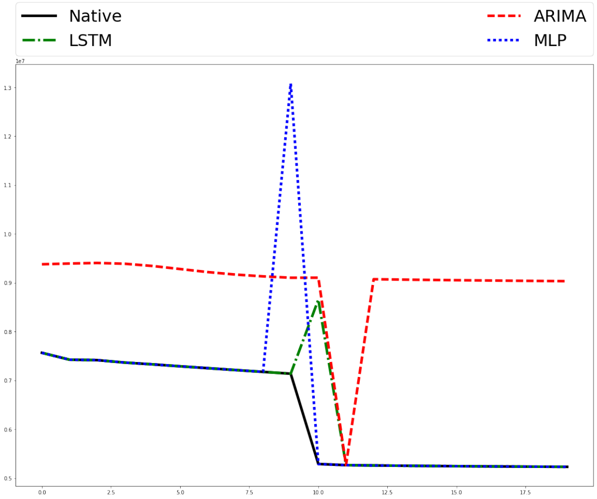


Figure 9. Summary routed flow characteristics

TABLE 1. Results: Absolute statistical quality

Quality	Native	ARIMA	LSTM	MLP
MEDIAN	6213078	9102802	7156210	6231733
MEAN	6283130	8978630	6451402	6580164
STD	1036927	862627	1130712	1805315
VAR (10^{12})	1.075	0.744	1.278	3.259
COUNT	1	2	1	1

TABLE 2. Results: Relative statistical quality

Quality	ARIMA	MLP	LTSM
MEDIAN	**1.4651**	1.151	1.003
MEAN	**1.429**	1.026	1.047
STD	**0.831**	1.090	1.741
VAR	**0.692**	1.189	3.031
COUNT	2	1	1

7. Conclusion

The results of simulation confirmed our approach to decrease standard deviation and variance of the dynamically routed overlay network is possible. The most usable method regarding the experiment is ARIMA. The experiment shows that proposed approach is feasible, but under specific condition and traffic characteristics using ARIMA method. In the following research, we will focus on the deep investigation of pro-active routing based on time series prediction using ARIMA and NN method. We are planning to implement new algorithms appropriate for overlay network routing based on traffic prediction. The next step is to explore the feasibility of used methods based on traffic characteristics. We will explore side effects of our presented approach on real-life audio/video streaming scenarios and large-scale applications.

Acknowledgments

The authors would like to thanks university specific research project SGS-2016-013 Advanced Graphical and Computing Systems.

References

[1] J. C. Granda, P. Nuo, F. J. Surez, D. F. Garca. Overlay network based on WebRTC for interactive multimedia communications. In *2015 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5, July 2015.

[2] A. Babay, C. Danilov, J. Lane, M. Miskin-Amir, D. Obenshain, J. Schultz, J. Stanton, T. Tantillo, Y. Amir. Structured overlay networks for a new generation of internet services. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1771–1779, June 2017.

[3] D. Ruppert and D. S Matteson. Time series models: Further topics. In *Statistics and Data Analysis for Financial Engineering*, pages 361–404. Springer, 2015.

[4] C. Katris and S. Daskalaki. Comparing forecasting approaches for internet traffic. *Expert Systems with Applications*, 42(21):8172 – 8183, 2015.

[5] I. Loumiotis, E. Adamopoulou, K. Demestichas, T. A. Stamatiadi; M. E. Theologou. On trade-off between computational efficiency and prediction accuracy in bandwidth traffic estimation. *Electronics Letters*, 50(10):754–756, 2014.

[6] H. Zare Moayedi and M. A. Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *2008 International Symposium on Information Technology*, volume 4, pages 1–6, Aug 2008.

[7] H. Feng and Y. Shu. Study on network traffic prediction techniques. In *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, volume 2, pages 1041–1044, Sept 2005.

[8] G. Feng. Network traffic prediction based on neural network. In *2015 International Conference on Intelligent Transportation, Big Data and Smart City*, pages 527–530, Dec 2015.

[9] X. Wang, C. Zhang, and S. Zhang. Modified elman neural network and its application to network traffic prediction. In *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, volume 02, pages 629–633, Oct 2012.

[10] A. Azzouni and G. Pujolle. A long short-term memory recurrent neural network framework for network traffic matrix prediction. *CoRR*, abs/1705.05690, 2017.

[11] H. Feng, Y. Shu, S. Wang, M. Ma Svm-based models for predicting wlan traffic. In *2006 IEEE International Conference on Communications*, volume 2, pages 597–602, June 2006.

[12] A. Y. Nikravesh, S. A. Ajila, C.-H. Lung, W. Ding Mobile network traffic prediction using mlp, mlpwd, and svm. In *2016 IEEE International Congress on Big Data (BigData Congress)*, pages 402–409, June 2016.

[13] R Sivakumar, E Ashok Kumar, and G Sivaradje. Prediction of traffic load in wireless network using time series model. In *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pages 1–6. IEEE, 2011.

[14] J. KACÁLEK and I. MÍČA. Nelineární analýza a predikce sít'ového provozu. (Nonlinear analysis and network traffic prediction) *VUT v Brně, Elektrorevue*, 2009.

[15] F. Hai-Liang, C. Di, L. Qing-Jia, C. Chun-Xiao Multi-scale network traffic prediction using a two-stage neural network combined model. In *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–5, Sept 2006.

Calculation of Cross-Sections of Boundary Surface Using Parallelization

Branislav Sobota
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovak Republic
branislav.sobota@tuke.sk

Milan Guzan
Dept. of Theoretical and Industrial Electrical Engineering
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovak Republic
milan.guzan@tuke.sk

***Abstract*— The paper presents the parallelization of calculations of cut through the boundary surface of the Chua circuit. The calculation itself implies a high demands for the computer processor and it is therefore referred to the technique of using parallelization in this case. We also point to the undisputed advantage of parallelization the boundary surface calculation. It is not only a much faster calculation on the new Intel i7 7700 processor, but also electric energy saving. Faster results also imply a better applicability of this circuit.**

***Keywords*— Chua's circuit, boundary surface, parallelization**

I. Introduction

The term chaos is used in dynamic systems whose state in time is else deterministic, but because of their complexity it is difficult predictable, irregular and non-periodic. A typical feature of chaotic systems is that a small change in their initial conditions can cause large changes in the dynamics of the analyzed system. Chua´s circuit was the first physical circuit that exhibits non-periodic (chaotic) behavior. The author of this circuit is prof. Leon Chua, who introduced it in 1983. Since then it has become one of the most studied non-linear electrical circuit [1], [2], [3]. The circuit is used for investigation chaotic phenomena, but its utilization is also in signal encryption or music [4], [5].

An important term in sequential and chaos-generating circuits is so-called boundary surface (BS). BS separates not only the attractors from each other but also indicates the size of regions of attractivity (RAs) corresponding to individual attractors. These are decisive for [6], [7]:

- explanation of multiple-valued logic memory failure,
- determining the magnitude of the control pulse when rewriting information in memory,
- decision about the maximum amplitude of the input signal that we want to encode with a chaotic signal.

BS cannot be measured. The only answer to the question: What is the size and morphology of BS?, can be provided only by simulation. The calculation of a single cut using only one CPU represents a relatively long time span [8], thus parallelizing the calculation of a large set of BS cuts represents a significant advance in the speed of the BS calculation as a whole [9] and then also in reconstruction the BS in 3D [10].

II. Chua`s Circuit and Boundary Surface

The Chua's circuit is shown in Fig. 1a. It is a simple circuit consisting of a non-linear element characterized by a negative differential resistance (NDR). IV characteristic of an element with NDR is shown in Fig. 1b. Chua's circuit is described by the system (1) and the mathematical expression of the element with the NDR is shown in (2). Parameter ρ in (1) corresponds to coil resistance.

$$\begin{aligned} C_1\left(\frac{du_1}{dt}\right) &= \frac{1}{R}(u_2-u_1)-g(u_1) \\ C_2\left(\frac{du_2}{dt}\right) &= \frac{1}{R}(u_1-u_2)+i \\ L\left(\frac{di}{dt}\right) &= -u_2 \end{aligned} \tag{1}$$

$$\begin{aligned} g(u_1) = m_2 u_1 &+ \frac{1}{2}(m_1-m_0).(|u_1-B_P|-|u_1+B_P|)+ \\ &+\frac{1}{2}(m_2-m_1).(|u_1-B_0|-|u_1+B_0|) \end{aligned} \tag{2}$$

Depending on the initial conditions (IC), the state of the circuit, with the parameters (3) can lead to a chaotic attractor (CHA) or a stable limit cycle (SLC).

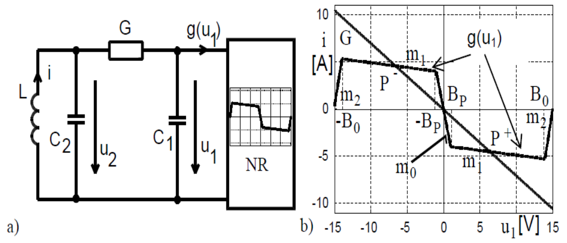


Fig. 1. a) Chua`s circuit, b) IV characteritics $g(u_1)$ with load G.

C_1=0,11111 F, C_2=1 F, $R=1/G$=1.42857 Ω, $\rho=0$ Ω, B_P=1 V, $L= 0.142857$ H, B_0=900 V, m_0=-0,8 S, m_1=-0,5 S, m_2=5 S (3)

One cross-section of the BS for parameters (3) is illustrated by Fig. 2. In this figure can be seen colored RA for the chaotic attractor (red) and also stable limiting cycle (yellow). In practice it means that if the IC (when the electrical circuit is connected to the supply voltage) lies in yellow or red RA, representative point (RP) will be attracted to SLC or CHA. Projection of CHA into plane i,u_1 illustrates Fig. 3.

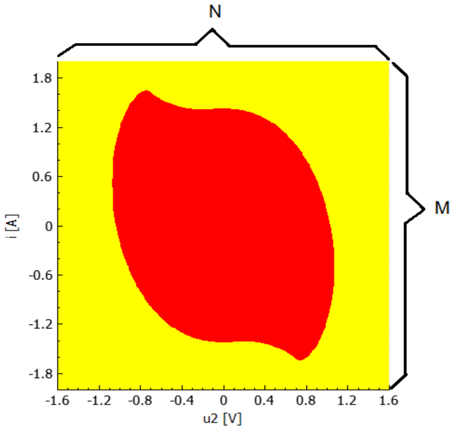


Fig. 2. The cut of the BS for $u_1 = 0$V.

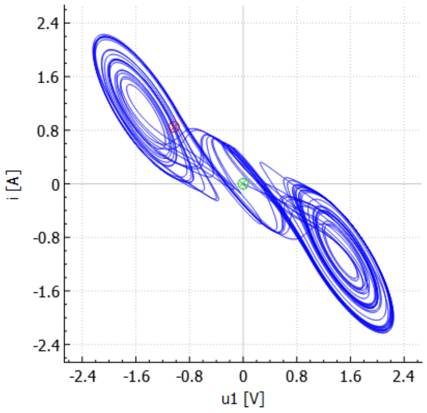


Fig. 3. Chaotic attractor for parameters (3) projectes in plane i,u_1.

If we calculate a large set of (100-300) cross-sections of BS, BS can be reconstructed in 3D [10], [11]. Thus, it is possible to make a coherent idea of the size and morphology of BS basically in a single image.

III. Principle of BS calculation and possibility of parallelization

Fig. 2 consists of $M\times N$ (400×400) different IC at u_1=0. When calculating BS, it is necessary to divide one plane into $M\times N$ IC. It is therefore necessary to calculate 160 000 trajectories across the whole plane. Each RP is attracted to one of the attractors from the corresponding IC, whereby attracting of RP is not dependent on the calculation of the previous or next trajectory. It means that there is possible to independently parallel calculate up to $M\times N$ trajectories! The principle of calculation one cross section is shown in Fig. 4 [9].

The process of the calculation of the BS started with the 0th column, where gradually 400 points (ICs) have been computed. After the completion of the last column in the 0th row, the process started over using the 1st row until the 400th row.

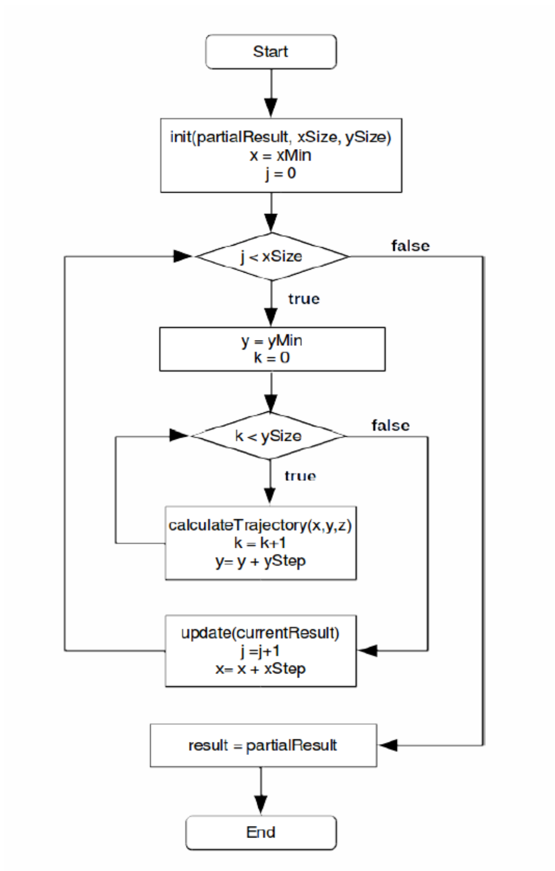


Fig. 4. Algorhitm for sequential calculation of a BS.

At the same time, the requirement to calculate the BS cuts is that the executable file of the designed program is fast. C++ language is very good in this case. Because the code written in C++ translates directly into the machine code, it provides high performance and many optimization options. It is also multiplatform and there are a large number of free libraries. For example the Qt software framework is good for implementation of the program graphical interface. Also Intel TBB is a software library that allows you to create parallel programs portable between computers. It provides APIs at a higher abstraction level than standard operating system threads for the programmer. The library allows you to define the tasks distributed dynamically among available processor cores. Job allocation is managed by the Task Scheduler component. It can use the Task stealing mechanism to compensate of the unequal load between the threads [8]. Intel TBB also provides several predefined generic parallel algorithms e.g.:

parallel_for - Allows simple parallelization for cycles with independent calculations in individual iterations.

parallel_reduce - Parallel calculation of one value based on multiple elements (e.g., a sum of numbers)

parallel_deterministic_reduc - Similar to the parallel_ _reduce algorithm, with the difference that the calculation is always guaranteed to be divided into the same number of sub-calculations.

parallel_scan - Parallel algorithm implementation Prefix sum

parallel_do - Parallelization of cycles in which the number of iterations is not known in advance.

parallel_pipeline - Allows you to apply more features to elements that some features can be executed parallel and some not.

parallel_foreach - Applies one function to each field element

parallel_sort - Allows parallel sorting of elements

parallel_invoke - Parallel call of multiple arbitrary void functions

The number of trajectories, we need to calculate to achieve the result when calculating the BS cuts, is always known at the start of the calculation. Individual calculations are independent of each other. For the parallel calculation of the BS cuts implementstion was the *parallel_for* algorithm as the ideal candidate. It is because the sequential version of the algorithm in Fig. 4 was realized using two different cycles *for*.

The *parallel_for* algorithm works in the following way. First, it is generally necessary to write how to calculate the for loop for any interval. Then enter the entire interval we want to calculate. Intel TBB automatically splits the interval specified and the jobs are distributed among the threads. The interval division algorithm is shown in Fig. 5. The default granularity of the calculation for *parallel_for* is 1. This means that each cycle iteration is considered as special task. According to [12] and [13], the creation of fewer tasks means lower overheads.

```
parallel_for (blocked_range<int>(Data, Data + N, GrainSize), Body ());
```

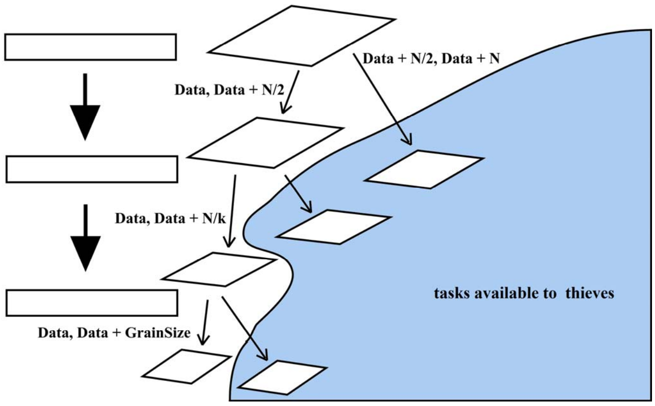


Fig. 5. Algorhitm of interval dividing in *parallel_for*.

But according to experiments in [14], too low a tasks number prevents the efficient division of tasks and usage of the task stealing mechanism. So the number of tasks should be several times higher than the number of cores to effectively usage of task stealing. But the individual tasks should be time-long enough to have the ratio of overheads to real calculations minimal. These conditions are met by our BS calculation. Based on these facts, it was decided that the outer for the cycle would be parallelized. So a calculation of one cut would be one task. This means: when the $M \times N$ cuts are calculated (where N equals the number of columns and M equals the number of rows) then N jobs will be created. In the worst case - if there is just one task left before the calculation is complete, it is possible to compute this task without parallelization, and only 1 processor core will be used in $1/M$ of the whole calculation. In practice, the size of the cuts is determined in hundreds of pixels, with a 400×400 cut size. It is a relatively small risk that the $1/400$ of calculation will not be parallelizable.

Thread stealing works as is illustrated in Fig. 6. When starting the calculation, the c threads will be created. The c is the default number of available CPU cores. In the figure we can see the case when the number of jobs $n=2c$ (at the beginning each thread has two tasks assigned). Let's assume that task 1 takes a considerably longer time than other tasks.

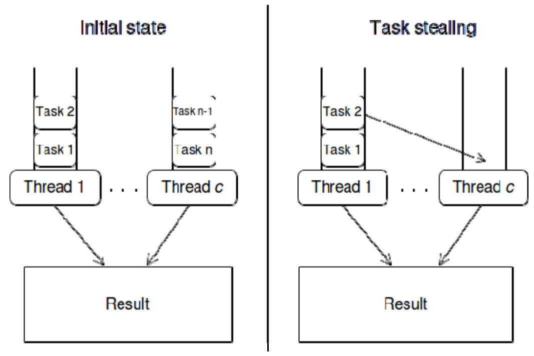
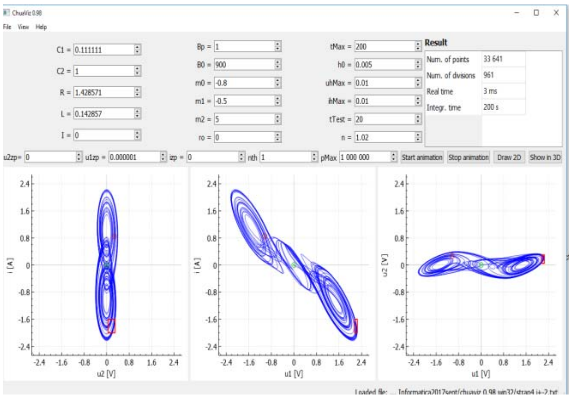


Fig. 6. The division of tasks between the threads and *Task stealing*.

On the right side of the diagram in Fig. 6 we can see a similar situation. The second thread has completed all of its tasks, but the first one still has two of them. In this case, the task stealing mechanism acts and the second task of the first thread moves into the thread 2. Thus, the thread 2 remains idle and the parallelization

IV. Implementation of Application

In collaboration with the author of the work [14], the application for BS calculation was designed. It is also necessary to calculate the trajectory and design a cuboid that would ensure the termination of the trajectory calculation corresponding to the CHA. Example of the first window - calculation and visualization of the trajectory, is illustrated in Fig. 7.



C1 = 0.111111
C2 = 1
R = 1.428571
L = 0.142857
I = 0
Bp = 1
B0 = 900
m0 = -0.8
m1 = -0.5
m2 = 5
ro = 0
tMax = 200
h0 = 0.005
uhMax = 0.01
ihMax = 0.01
tTest = 20
n = 1.02

Result

Num. of points	33 641
Num. of divisions	961
Real time	3 ms
Integr. time	200 s

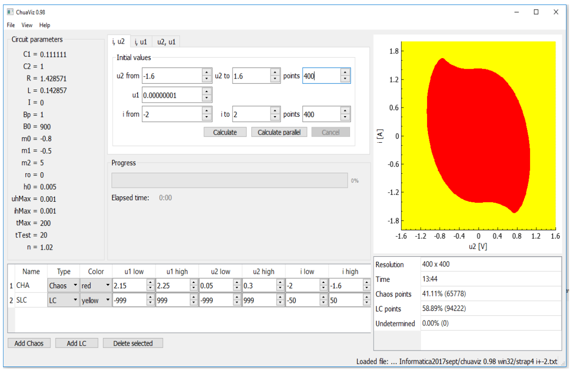
u2zp= 0 u1zp = 0.000001 izp = 0 nth 1 pMax 1 000 000

Loaded file: ... Informatica2017sent/chuaviz 0.98 win32/stran4 i+-2.txt

Fig. 7. The application window after calculation and visualization of trajectory.

Parameters (3) and other parameters and the trajectory calculation are specified at the top of the window. At the bottom there are 3 projections of trajectory in the order (from left to right): i,u_2; i,u_1 and u_2,u_1. The red rectangle at the edge of the CHA represents the mentioned cuboid. When the RP enters it, the calculation stops, the code for the CHA is assigned to corresponding IC and new IC is read.

The size of the cuboid is evident from the bottom of Fig. 8. Here are in the row:1 CHA the indicated dimensions in the limits u_2, u_1, i. The row:2 SLC is used for test calculation of the termination of SLC. The calculation stops if the RP during the calculation exceeds any of the limits u_2, u_1 and i.. The top of the window is divided into three columns (from left to right).



Circuit parameters

C1 = 0.111111
C2 = 1
R = 1.428571
L = 0.142857
I = 0
Bp = 1
B0 = 900
m0 = -0.8
m1 = -0.5
m2 = 5
ro = 0
h0 = 0.005
uhMax = 0.001
ihMax = 0.001
tMax = 200
tTest = 20
n = 1.02

Initial values

u2 from -1.6 u2 to 1.6 points 400
u1 0.00000001
i from -2 i to 2 points 400

Progress 0%

Elapsed time: 0:00

	Name	Type	Color	u1 low	u1 high	u2 low	u2 high	i low	i high
1	CHA	Chaos	red	2.15	2.25	0.05	0.3	-2	-1.6
2	SLC	LC	yellow	-999	999	-999	999	-50	50

Resolution	400 x 400
Time	13:44
Chaos points	41.11% (65778)
LC points	58.89% (94222)
Undetermined	0.00% (0)

Loaded file: ... Informatica2017sept/chuaviz 0.98 win32/strap4 i+-2.txt

Fig. 8. The application window for computation of one BS cut.

The parameters (edited in the previous window in Fig. 7) are visible in the first column. The parameters for the calculation of one cut in planes, i,u_2; i,u_1 and u_2,u_1 and the number of points M×N are in the middle column,. The BS cut is computed and visualized in the last (right) column. This is the cut illustrated in Fig. 2. The resolution, the computing time needed to calculate of 1 cut (Time), the number of points and the percentage from 100% belong to the red or yellow RA (*Chaos points* or *LC points*) are displayed In the table below. The last *Undetermined* displays the number of ICs that were not attracted to any of the attractors. In this case, it is 0, i.e., all ICs were attracted to some of the attractors. The view to calculation progress of one BS by using of the parallelization is in Figure 9 (black bars are the not calculated areas yet).

V. The Comparison of Parallel Calculations Performance by Using Only 1 CPU

As is shown in Fig. 8, the calculation time of 1 cut was 13:46 min. The computational time is relatively short. The calculation was performed on a modern i7 7700 4.2GHz processor and 48GB RAM, 4cores / 8 threads (4). Not every computer is so equipped, then the comparing of the same section calculation speed is illustrated in Table 1. It also includes a speed comparison with using only 1 CPU. As the Table 1 shows calculation of 1 BS using normal PC2 - PC5 with only 1 CPU takes several hours. And since modern computers usually have 4 threads, their use is more than adequate. The fastest calculation of 1 BS is, of course, by using PC1 - a computer at a price of 1 100 €. Of course, we can expect further increases in parallelization performance with each new processor in the future. It would be interesting to test the latest processors from AMD (Ryzen) or Intel (Intel Core i9 or Intel Core i9 Extreme Edition).

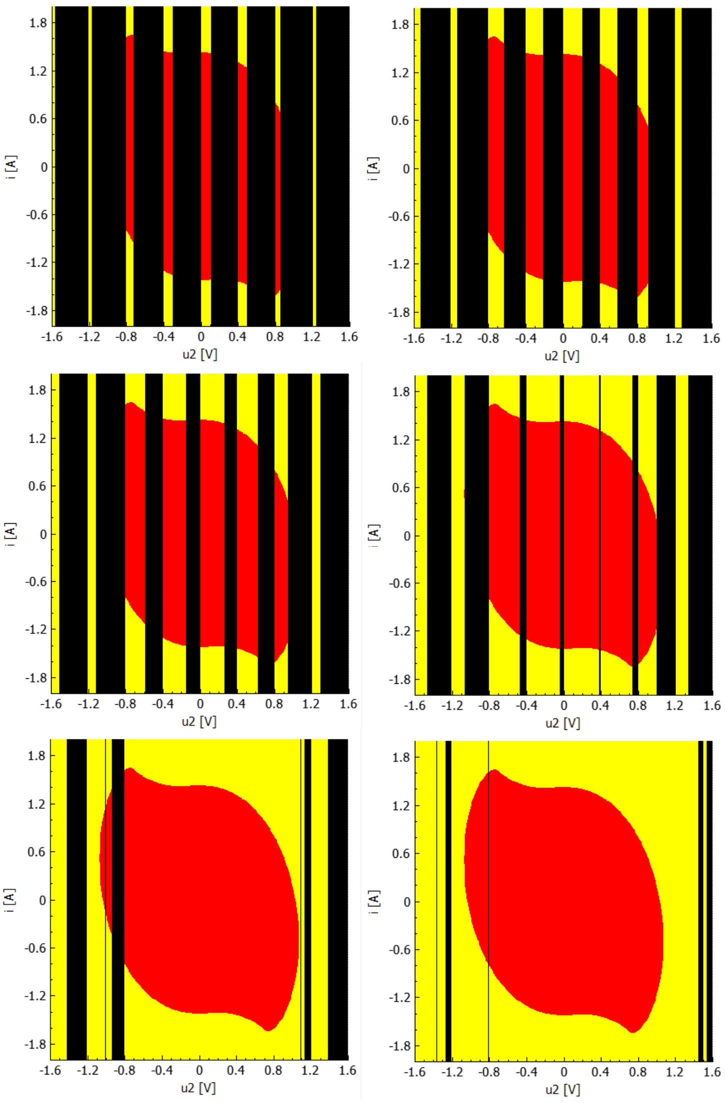


Fig. 9. Calculation procedure by using 8 cores with parameters (4).

TABLE I. Calculation Speed of 1 Cut BS Using 1 CPU and Paralelisation

Computers	1CPU [min:s.]	Paralel computing [min:s]
PC1	86:34	13:46
PC2	139:43	29:19
PC3	141:27	49:15
PC4	369:28	133:22
PC5	274:58	136:16

TABLE II. Parameters of Individual Computers

Name	CPU model	OS	cores / threads
PC1	Intel i7-7700, 4.2 GHz	Win 10 64 bit	4 / 8
PC2	Intel i7-3770, 3.4 GHz	Win 7 64 bit	4 / 8
PC3	Intel i5-5200U, 2.2 GHz	Win 8.1 64 bit	2 / 4
PC4	Intel E5300, 2.6 GHz	Win 10 32 bit	2 / 2
PC5	AMD Athlon II X2 220, 2.8 GHz	Win 10 32 bit	2 / 2

VI. Conclusion

Although modern computers have more than 2 threads - a standard user does not use more than one computer core. Multiple cores can be used in video, graphics, or virtual reality applications. However, the standardization also should be in the parallelization of calculations in applications that deal with systems (partial) of differential equations. It does not matter whether it is an analysis of chaos-generating circuit or multiple-valued memory [16] - [19]. The computational difficulty of the problem determines a parallelization. As is shown in Table 1, the calculation of BS cuts is such a difficult case and therefore we have pointed out in this paper the possibility of parallelization such and similar calculations, concluding that not only the more processor threads but also the newer processors are needed to speed up the calculation. The PC7 with the i7 7700 processor is definitely the fastest. If a few such PCs should be used, it is not only about quicker calculation but also about multiple saving of the company's funds for electricity consumption. It could also, in this case, help to configure the parallelization and its granularity directly from the application menu, which is not implemented yet (only in the source code). Such changes could lead to several attempts, and they may, in some cases, lead to better results.

Interestingly, the performance of 1 CPU for PC3 is equal with PC2, although PC3 CPU frequency is lower (1.2 GHz) compared to PC2. It is a fifth- respectively third-generation of Intel processors and PC3 is a notebook. PC 2 for parallelization is faster because it has 4 threads (more than PC3). Another interest was shown when comparing performance using a single processor (in Table 1, column 1CPU). PC4 is slower than PC5, although parallel calculations are approximately equal. This test only confirmed the hint from work [9], where PC4 time - 3:55 min. and PC5 time - 2:54 min are presented for similar processors (PC4 - Intel E5200 2.5 GHz Win 7 32 bit 2/2, PC5 - AMD Athlon II 215, 2.7 GHz Win 7 32 bit 2 / 2) in Table III. These CPUs were sold in PCs when CPU performance was for 1 CPU better for AMD processors than Intel. This comparison also shows the benefits of investing to the latest processors from Intel. It is still a question: how the most recent and especially cheaper processor AMD Ryzen would have compared to our calculations.

Because chaos is also used in rehabilitation or in applications for some disabled people when the patient/person can not predict the rehabilitation/assistance movement of the device, it is appropriate to deal with it. E.g. the principle of a rehabilitative shoe whose motors are controlled by CHA was described in [20]. But not every CHA is usable in the field of rehabilitation/assistance to disabled people. The result is not only the understanding of the behavior of such circuits, but also the choice of the appropriate CHA type for further use in the area of rehabilitation or assistance technologies for disabled people.

Acknowledgment

This work has been supported by the KEGA grant no. 083TUKE-4/2015 „Virtual-reality technologies in the process of handicapped persons education.“

References

[1] L. O Chua., “The Genesis of Chua’s circuit,” AEU, Vol. 46, 1992, No. 4, pp 250–257.

[2] Chua’s Circuit – A Universal Paradigm for Generating and Studying Chaos. Prem Bhushan Mital1, Umesh Kumar2, and Rai Sachindra Prasad1. J. of Active and Passive Electronic Devices, Vol. 3, pp. 51–63. Old City Publishing, Inc. 2008.

[3] http://people.eecs.berkeley.edu/~chua/circuitrefs.html

[4] O. Vasilovici et al., “Signal Encryption Using Chua’s Chaotic Circuits. Journal of Advanced Research, ” in Physics 4(1), 011307 (2013).

[5] G. Kress, I. Choi, N. Weber, R. Bargar, A. Hubner, “Musical signals from Chuas circuit, ” IEEE Transaction on Circuits and Systems – II, vol. 40, no. 10, oct. 1993

[6] P. Galajda, M. Guzan, V. Špány, “The Control of a Memory Cell with the Multiple Stable States, Radioelektronika 2011 : April 2011, Brno, Brno University of Technology, 2011 pp. 211-214.

[7] M. Guzan, P. Galajda, L. Pivka, V. Špány, “Element of singularity is a key to laws of chaos,” Radioelektronika 2005, Brno University of Technology, 2005 pp. 33–36.

[8] M. Guzan, B. Sobota, J. Astaloš, “Výpočet hraničnej plochy použitím technológií Grid a GPGPU, ” Posterus.sk. vol.. 5, no. 10 (2012), pp. 1-9.

[9] Zs. Rácz, M. Guzan, B. Sobota, “Parallelizing boundary surface computation of Chua’s circuit,” Radioelektronika 2017, april 2017, Brno, pp. 1–4.

[10] B. Sobota, “3D modelling of Chua’s circuit boundary surface,” Acta Electrotechnica et Informatica, vol. 11, no. 1, pp. 44–47, 2011, ISSN 1335-8243.

[11] M. Guzan, B. Sobota, “Vizualizácia stavového priestoru nelineárnych autonómnych obvodov, ” ISBN 978-80-87321-31-7. JUPOS. 2015.

[12] Intel® Software – Algorithms. https://software.intel.com/en-us/node/506140

[13] G. Contreras, M. Martonosi, “Characterizing and Improving the Performance of Intel Threading Building Blocks,” Department of Electrical Engineering Princeton University.

[14] K. Farnham, “TBB Parallel_for Grain Size Experiments,” https://software.intel.com/en-us/blogs/2007/08/07/tbb-parallel_for-grain-size-experiments.

[15] K. Farnham, “Hierarchically Tiled Arrays and Threading Building Blocks.https://software.intel.com/en-us/blogs/2007/09/28/hierarchically-tiled-arrays-and-threading-building-blocks

[16] V. Špány, P. Galajda, and M. Guzan, “The state space mystery in multiplevalued logic circuit with load plane – part I,” Acta Electrotechnica et Informatica, vol. 1, pp. 17–22, 2001.

[17] T. Gotthans and J. Petržela, “Experimental study of the sampled labyrinth chaos,” Radioengineering, vol. 20, no. 4, pp. 420–427, 2011.

[18] M. Guzan, “Limit cycles and trajectories in multilevalued memory cell,” Acta Electrotechnica et Informatica, vol. 2, no. 1 (2002), pp. 36–41, (in slovak.)

[19] M. Guzan, “Boundary surface of a ternary memory in the absence of limit cycles,” Radioelektronika, Proc. of 22-nd international conference: April 17-18, 2012, Brno, pp. 1–4, 2012.

[20] Šimšík, D.; Galajdová, A., Drutarovský, M.; Galajda, P.; Pavlov, P., “Wearable non-invasive computer controlled system for improving of seniors gait, “ International Journal of Rehabilitation Research, vol. 32, no. 1, p. 35–38, 2009.

Usage of Optical Correlator in Video Surveillance System for Abandoned Luggage

Dávid Solus
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
david.solus@tuke.sk

Ľuboš Ovseník, Ján Turán
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
lubos.ovsenik@tuke.sk; jan.turan@tuke.sk

***Abstract*—This article is focused on new methods of image processing used in modern video surveillance systems. The aim of this article is usage of the latest knowledge in the field of digital signal processing (image and video) and optical correlation in the design and experimental verification of video surveillance system that should detect lost or abandoned luggage.**

***Index Terms*—Image processing, optical correlator, video surveillance system**

I. Introduction

Nowadays, the evolution of security is even faster with help of science and new technology. A significant development in computer technology in recent decades also reflected in the development of safety systems. A simple camera system linked to multiple displays allows security officer to monitor the situation in the monitored area in real time is now replaced by a much more comprehensive solution. The analog signal is replaced by digital one and it is generally treated by control server. The software suite enables intelligent monitoring of the area, face recognition of unauthorized persons or detection of illegal activity. Algorithms used in such applications generally use methods and techniques of digital signal processing and optical correlation [1, 2].

Our goal was to create a test system for detecting abandoned luggage, which would employ above mentioned technologies. The system can detect objects in the input image (person, luggage) and track their movement (by monitoring their coordinates).

II. Design of Video Surveillance System for Abandoned Luggage with Cambridge Optical Correlator

One of the possible methods for object recognition in video surveillance systems used in security applications can also be optical correlation. The aim of these applications is mainly to support and facilitate the work of security personnel in various buildings such as airports, subway, banks, casinos etc. Their task is to detect presence and movement of people and things, recognize and classify patterns in video footage. Subsequent analysis of acquired data can alert staff to potential threats (Threat Detection), the presence of unauthorized persons (Intrusion Detection) etc.

This designed and tested system consists of two main parts, hardware and software part. The hardware portion consists of following blocks:

- camera system,
- optical correlator,
- server (Fig. 1).

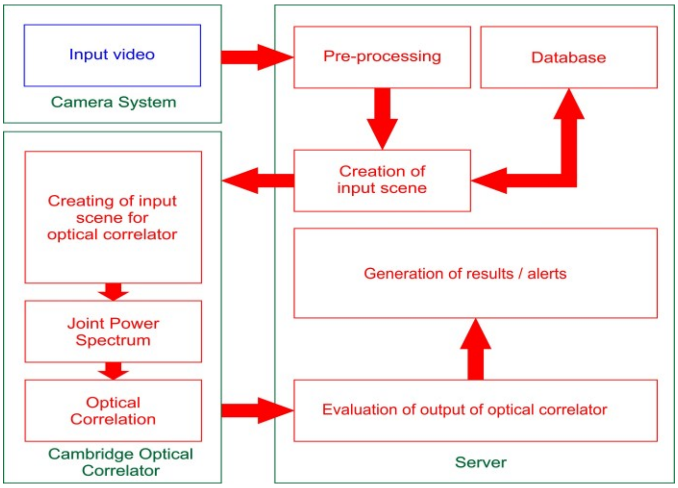

Input video
Camera System
Pre-processing
Database
Creating of input scene for optical correlator
Creation of input scene
Generation of results / alerts
Joint Power Spectrum
Optical Correlation
Evaluation of output of optical correlator
Cambridge Optical Correlator
Server

Fig. 1. Block diagram of video surveillance system.

The input video is captured by the camera system, consisting of one or more IP cameras connected via network infrastructure to other elements of the system. Server provides computing resources for the software part of this system. It consists (depending on the purpose of the system) of the following blocks:

- pre-processing of the input video,
- database (contains reference samples, resp. models used for correlation),
- preparation of the input image before the optical correlation and communication with the optical correlator (using the API - Application Programming Interface),
- evaluation of the output of the optical correlator,

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

- results generating, resp. alerts (depending on what the video control system is used).

Cambridge optical correlator belongs to a group of JTC also referred to as 1/f Phase-Only Joint Transform Correlator. The expression 1/f indicates only a simple Optical Fourier Transform in both stages of the process JTC. This means using the same optics twice in the process of correlation.

Block pre-processing includes various methods of digital image processing (gamma correction, color filtration, edge detection, etc.) to ensure better quality of the input of the optical correlator. The result of correlation is obtained values of similarity, respectively match of the image captured by the camera system and the reference image from the database, as well as the coordinates of objects in the entrance plane [3, 4, 5, 6].

III. Experiments and Results

Basis of proposed video surveillance system is evaluation of the distance of objects detected in the monitored area. The sequence of input images is pre-processed by using applications designed and programmed in C#. Then they are analyzed by Cambridge optical correlator. Based on the coordinates of these objects we can determine whether the subject tries to leave the luggage.

A. Obtaining of experimental images

To simplify the experiment, input video was temporarily replaced by a sequence of images. The experiment was carried out with four sets of frames on which are generally three types of luggage and two people. Each set contains five frames and each frame consists of a luggage, which is still in the same position and the man who is gradually moving away from it (Fig. 2).

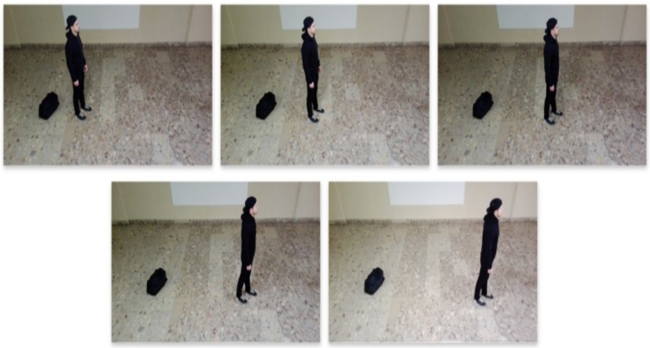


Fig. 2. Example of sequence of input images.

First, each image is pre-processed (Fig. 3) and it consists of:

- Median filter (Fig. 3a) - noise reduction of the input images,
- gamma correction (Fig. 3a) - adjustment of values of color components,
- color filtration (Fig. 3b) - detection of objects based on certain colors, in this case mainly black,
- converting the image into the grayscale (Fig. 3c),
- edge detection using Sobel operator (Fig. 3d),
- blob filtration (blob in this case is a white object on a black background) under the minimum size (Fig. 3e),
- filter to fill the remaining objects in white (Fig. 3f).

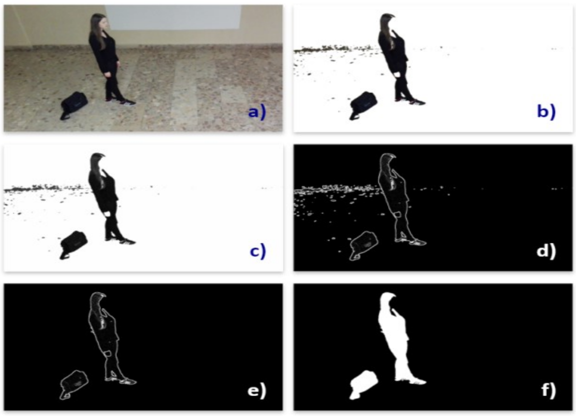


Fig. 3. Pre-processing of input frame.

The result of the pre-processing by using several filters (gamma correction, color filter, edge detection, etc.), and functions is the image that is suitable for further processing by the optical correlator. All images are stored in the destination directory from which they are the subsequently-loaded into Fourier Optics Experimenter. The sequence of images after pre-processing is shown in Fig. 4.

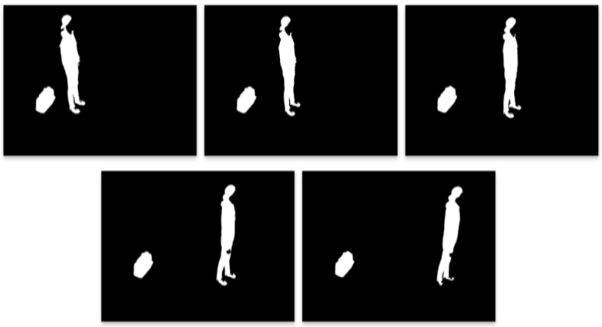


Fig. 4. Sequence of input frames after pre-processing.

Optical correlation is then performed between the first frame and the first frame and the first frame and any other frames. In terms of design, the most important data are the changes of coordinates of objects between frames. Based on these data the system should detect abandoned baggage, resp. a person who tries to abandon this baggage.

B. Evaluation of experiments

Cambridge optical correlator has an input and reference image aligned and displayed together on the Spatial Light Modulator. Subsequently, these images are transformed by Fourier transformation and a non–linear camera then captures the intensity distribution of transform to produce Joint Power Spectrum (JPS). JPS is then binary or threshold processed and this processed image enters to transform process as input image of second FT. Output of this transform is the correlation plane that includes correlation peaks per match.

Fig. 5 shows the change of the position of the correlation peaks in the output plane of the optical correlator based on the change in position of an object (white squares) in the entry level.

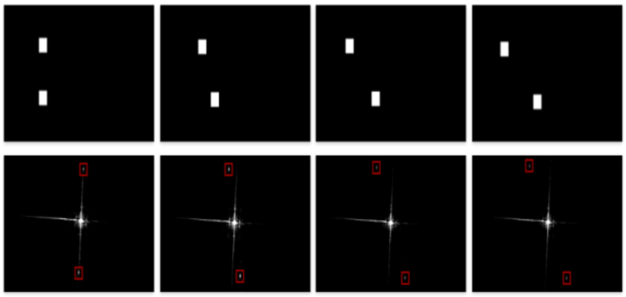


Fig. 5. Example of sequence of input images.

Obtained coordinates of correlation peaks are shown in TABLE I. The upper left corner of the output plane has coordinates [0, 0]. Fig. 5 and TABLE I. show that if you increase the distance of objects in the input plane in the direction of the X-axis, the distance of correlation peak at the output increases in the same direction (coordinate difference Δx). The same principle would apply also when to another position in the opposite direction, respectively the Y-axis direction.

TABLE I. COORDINATES OF CORRELATION PEAKS

Test	x_1	y_1	x_2	y_2	Δx
1.	257	213	243	384	14
2.	235	210	265	386	30
3.	213	207	288	389	76
4.	201	206	299	390	98

The process of optical correlation (Fig. 6) consists of the following steps:

- loading of images into the input plane (Fig. 6a),
- selection of the ROI to obtain spectrum JPS (Fig. 6b),
- setting of parameters of CMOS camera
- selection of the ROI of correlation peaks (Fig. 6d),
- obtaining the coordinates of the correlation peaks.

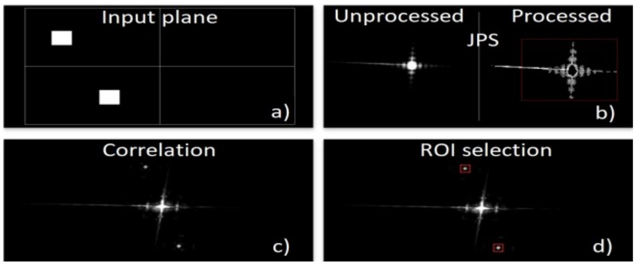


Fig. 6. Process of optical correlation.

The principle of obtaining the coordinates of the object is used in the design of video surveillance systems for abandoned luggage. Based on these coordinates it is also possible to evaluate whether the objects are closer to or any from each other. So, the system is able to warn whether a person is attempting to leave luggage in the monitored area.

Fig. 7 shows the change of the position of the correlation peaks in the output of the optical correlator (ROIs are highlighted in red). The figures show that if you change the position of objects in the input plane, position of the correlation peaks in the output plane of correlator is also changed. The coordinates of these peaks are obtained by analyzer. Their comparison enables tracking the movement of objects in the monitored area, and also enables measuring the distance between them. In the case that the monitored person is leaving from luggage in the opposite direction, position of correlation peak was mirror-inverted.

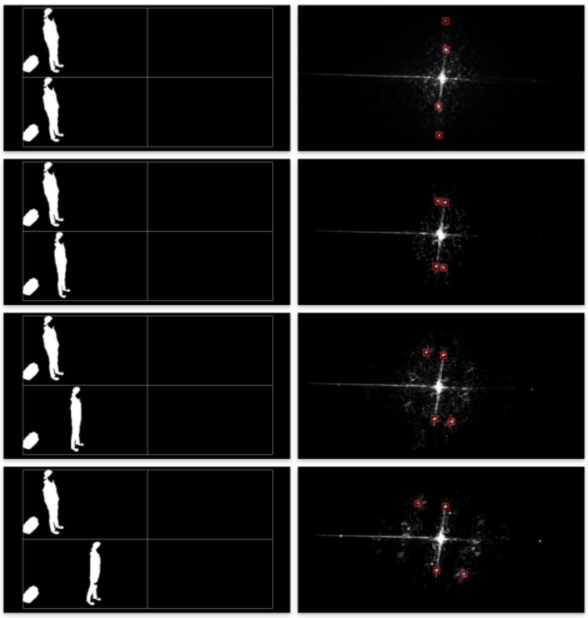


Fig. 7. Sequence of experimental frames into input of optical correlator (left) and output of optical correlator (right).

TABLE II. RESULTS OF EXPERIMENTS (1- 4)

Combination of Frames	Δx			
	Exp_1	Exp_2	Exp_3	Exp_4
1 – 1	7	7	7	7
1 – 2	7	7	8	14
1 – 3	27	25	31	34
1 – 4	49	45	52	49
1 - 5	67	70	77	71

The results of all four experiments are shown in TABLE II. The first column contains information about which two images were used in the correlation process. Other columns contain the calculated change of distance of peaks that represent luggage and silhouettes of people. Based on these values, video surveillance system evaluates movement of the monitored person. If the value of Δx has exceeded a predetermined value, the system can alert by generating a certain message type. Row 2 presents the results of the experiment (Exp_1) in Fig. 7. The results of the other three experiments (Exp_2 to Exp_4) are given in columns 3, 4 and 5.

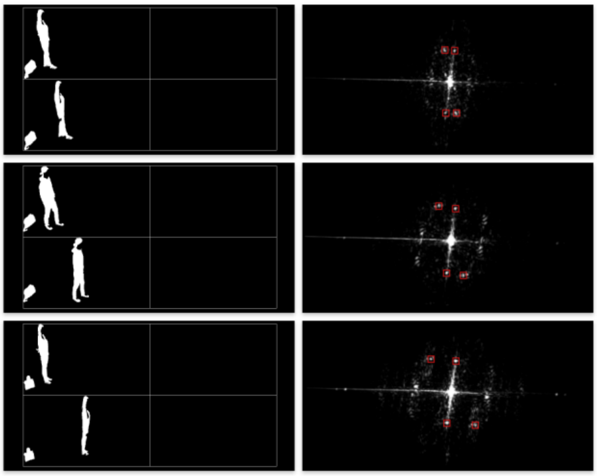


Fig. 8. Comparison of the input frames and outputs of correlator of other three experiments.

From comparison of obtained data, it can be seen that system in all four experiments recognizes the effort of people to leave luggage and move away from it. Gradually widening difference values Δx are present in each of experiments. In this way, it would be possible to obtain object location even if person tries to leave the area in a different direction. If it was the case that the correlator output will contain only two correlation peaks (correlation between luggage), the person probably completely left the monitored space [7, 8, 9, 10,11,12].

IV. Conclusion

The aim of the experiment was to use new methods in the field of digital image processing and optical correlation to design a simple video surveillance system for abandoned luggage. Its task was to detect and track the movement of objects (person, luggage) in the monitored area.

The proposed system was tested by performing the experiment, which consisted of the following elements:

- the creation of four sets of input frames that contains two persons and three kinds of luggage,
- image processing by the program created in C# - pre-processing of inputs before optical correlation,
- comparison of input frames by optical correlation,
- obtaining coordinates of objects in the monitoring area,
- evaluate changes of coordinates, and the movement of objects.

The coordinates represent the position of the correlation peaks that characterize the similarity of objects in the input scene of optical correlator. The cross-correlation of luggage between each frame was relatively high, so correlation peaks are much more evident than the correlation peaks between the silhouettes of people moving away from the luggage. The movement of people around the scene caused a fluctuation of the value, but after a slight adjustment of camera parameters we could obtain the coordinates of these correlation peaks. Based on the changes of coordinates obtained by experiments can be evaluated and track the movement of objects in the area. After exceeding a certain critical distance, we can evaluate that person tries to leave luggage.

Acknowledgment

This work was supported by Cultural and Educational Grant Agency (KEGA) of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the project no. „023TUKE-4/2017 - New Trends in the Optical Data Transmission “, and as support of the Scientific Grant Agency VEGA project no. „1/0772/17 - Multiple Person Localization Based on Detection of Their Vital Signs Using Short-Range UWB Sensors “.

References

[1] B. Cyganek, “Circular Road Signs Recognition with Soft Classifiers,” Integrated Computer-Aided Engineering, Vol. 14, No. 4, pp. 323-343, 2007.

[2] B. V. K. V. Kumar, A. Mahalanobisbis, R. D. Juday, “Correlation Pattern Recognition,” Cambridge University Press, ISBN 9780521571036, 2005.

[3] D. Soendroro, I. Supriana, “Traffic sign recognition with Colour-based Method, Shape-arc estimation on SVM,” in International Conference on Electrical Engineering and Informatics (ICEEI), pp. 1-6, 17-19 July 2011.

[4] C. Bahlmann, Y. Zhu, R. Visvanathan, M. Pellkofer, T. Koehler, “A system for traffic sign detection, tracking, and recognition using colour, shape, and motion information,” Proceedings. IEEE Intelligent Vehicles Symposium, 2005, pp. 255-260, 6-8 June 2005.

[5] Z. Song, B. Zhao, Z. Zhu, M. Wang, E. Mao, “Research on Recognition Methods for Traffic Signs,” FGCN '08. Second International Conference on Future Generation Communication and Networking, 2008, pp. 387-390, 13-15 Dec. 2008.

[6] S. Vitabile, A. Gentile, and F. Sorbello, “A neural network based automatic road sign recognizer,” presented at The 2002 Inter. Joint Conf. on Neural Networks, Honolulu, HI, USA, 2002.

[7] P. Ambs, “Optical Computing, A 60-Year Adventure,” in Advances in Optical Technologies Journal, 2010.

[8] Cambridge Correlators, http://www.cambridgecorrelators.com, May 2017.

[9] Cambridge Correlators, “Fourier Optics Experimenter,” User Guide.

[10] J. W. Goodman, Introduction to Fourier Optics, Colorado: Roberts & Company.

[11] P. Ivaniga, T. Ivaniga, “10 Gbps optical line using EDFA for long distance lines “, Przeglad Elektrotechniczny, vol. 93, no. 3, 2017, pp. 193-196.

[12] T. Ivaniga, P. Ivaniga, “Comparison of the optical amplifiers EDFA and SOA based on the BER and Q-factor in C-band”, In: Advances in Optical Technologies, doi:10.1155/2017/9053582, 2017,pp.1-9.

Quantum programming: a review

Milan Spišiak
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice
Letná 9, 042 00, Košice, Slovakia
Email: milan.spisiak@tuke.sk

Ján Kollár
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice
Letná 9, 042 00, Košice, Slovakia
Email: jan.kollar@tuke.sk

***Abstract*—In this paper, we discuss an actual state of the quantum programming. At present, we don't how know to solve some algorithmic problems, mainly the processing of complex information, using the current computer architecture in real time. The quantum approaches promise new opportunities in the processing of complex information. This paper is divided to three parts. The first part of paper is focused on the theoretical basis of quantum computer. The second part presents current state in this area. The third part shows experiments on a quantum computer. The experiments are performed on IBM Q platform.**

***Index Terms*—quantum programming, IBM, ibmqx2**

I. Introduction

Researchers have been trying to create an universal quantum computer for a commercial usage for the last few decades. The last few years were very important, because the level of quality of these computers was reached, but now we can test programs on real quantum computers. Quantum computers(QCs) can solve problems which are unsolved by present computers in an acceptable time. This is the reason why researchers have constructed a quantum computer. Quantum computers aren't effective to use for "basic" usage, such us administrative work or surfing the Internet. QCs are effective to solve some NP class problems, because they can process these problems in the polynomial time which classical computers are not able to do. They are able to effectively solve massive parallel issues as well.

The potential of quantum computers is in a lot of different areas, such as medicine, pharmacy or chemistry. We are able to simulate varying combinations of molecules and interactions between them at a massive scale. Computers based on fpga or gpu aren't able to compete in these specific situations. However, when we try to solve an issue with a lower level of parallelism, or a non-NP problem, the QCs lose a performance.

Quantum computers work on different principles than the usual computer. They use quantum theory. The way how we create our programs for these computers is therefore different. The two attributes of these computers make their architecture so exciting. The first is a qubit and the second is a quantum entanglement. Both attributes stand for a fundamental base of these computers. However, their characteristics are very strange against the standard computer principles. Algorithms are therefore completely different. This paper will present these fundamental principles and will also show some quantum algorithms.

The aim of the paper is to present the current state in this area of research and present some of the performed experiments on the IBM Q platform.

II. Fundamental principles

To understand quantum algorithms we need to know the fundamental terms and principles of QCs at first. This section is focused on this aim.

A. Definition Of Qubit

The fundamental data unit in classical computer is a bit. As we know the bit can be 2 values, either 0 or 1. The fundamental data unit in the quantum computer is a quantum bit - a qubit. Valiron's [1], Smith's [2], Dong's [3] works define the qubit as a ray in a 2-dimensional Hilbert space. Palet [4] subsequently explains that values which are similar to the classical values of 0 or 1 are denoted as $|0\rangle$ and $|1\rangle$. The qubit also contains another state which stands for a combination of both states simultaneously. This state is known as a quantum superposition. This state is described as $\psi = \alpha|0\rangle + \beta|1\rangle$. When we perform the measurement of the qubit state we always gain either 0 or 1. α and β represent the probability of occurrence of particular state in the result. That means we don't know the exact state of the qubit before the measurement. We are just able to say the probability of particular value in the result. A Hadamard gate is used to create the superposition. The Hadamard gate has one input argument. The input argument can be the bit or the qubit. The Hadamard gate result is the qubit.

A block sphere is an useful tool to visualize the state of the qubit as Nielsen describes in his work [5]. A lot of operations performed on the qubit can be described on the block sphere such as a Pauli's X gate. However, we have to keep in mind that the state of multi-qubit systems can not be visualized on the block sphere so straightforwardly. Fig.1 is an example of the block sphere. A measurement of the quantum state is performed on the z axis. Because of this, there are values $|0\rangle$ and $|1\rangle$. $|\psi\rangle$ is an actual position of the qubit and φ,θ are numbers defining a point on the three-dimensional sphere. As Nielson [5] says, the number of points in the sphere can be almost an infinite. He continues and explains the measurement

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

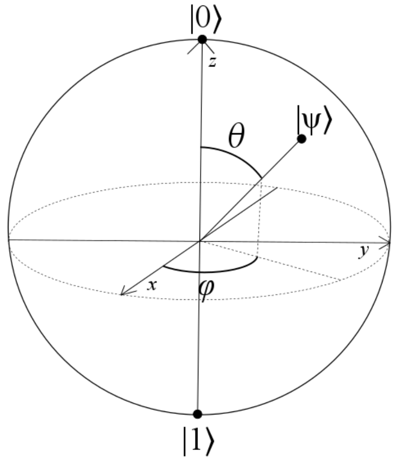


Fig. 1. Qubit block sphere representation

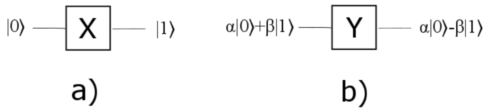


Fig. 2. a) quantum gate X, b) quantum gate Y

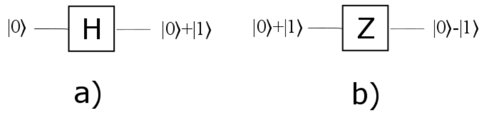


Fig. 3. a) quantum gate H(Hadamard), b) quantum gate z

process. When we perform the qubit measurement we always get either 0 or 1.

B. Quantum Entanglement

The quantum entanglement is the second important attribute of the quantum computer. The quantum entanglement is a special state of two quantum particles. If two particles are entangled, a change of state is applied on both particles. There are two possible scenarios of the entanglement. The first scenario sets up the second particle state to the same state as the first particle is in. The second scenario sets up the state of the second particle to the opposite state of the first particle. To create the quantum entanglement we can use an EPR circus. The EPR was named after the founders of the quantum entanglement principles: Einstein, Podolsky and Rosen [6]. The implementation of EPR circus, at the ibmqx2 platform, will be described in a next part of the paper.

C. Fundamental quantum circles

To create a quantum program we need tools allowing us to manage the quantum computer. For this aim we use quantum circles. Different quantum computers use different quantum circles. We therefore describe fundamental circles of ibmqx2 [7] in this paper. The ibmqx2 uses 8 fundamental circles: Pauli's X,Y,Z gates, a Hadamard gate, a Control NOT gate, a measurement gate and phase gates S and T.

The gate X is one of the elementary quantum gates (a scheme is shown on Fig.2a). This gate preforms a π-rotation about X-axis. If we visualize this operation on the block sphere we see that the state at the Z axis is changed to the opposite state. Because the ibmqx2 performs the measurement of qubit state based on Z basis, the gate flips the value to the opposite value. In another words it performs a NOT operation. The X gate is therefore called a value flip gate.

The gate Z changes the phase of the qubit. If we visualize this process on the block sphere we see that the Z gate performs a π-rotation about Z-axis. This process modifies the phase of the qubit. The phase influences the probability of the state occurrence, we mean states 0 or 1, in the result. If we want to change the probability to α = 25% and β = 75% we need to change the phase to $\pi * 0.25$. For this aim we have to use another gate allowing us to set up the specific value of the phase. The Z gate allows us to change the phase about π only.

The gate Y preforms a π- rotation around the Y-axis. Fig.2b shows a particular gate scheme. The X gate is the bit-flip gate, the Z gate is a phase-flip gate and Y gate is both. It changes the bit state and the phase state of qubit simultaneously.

The Hadamard gate (Fig.3a) creates the superposition state. What does it mean for us? This very extraordinary state of particles allows us to increase the computation potential of these class computers. The gate creates a particle with states 0 and 1 at the same time. The computation is therefore evaluated on both states instantly. The gate creates both values with 50% probability of occurrence in the result.

As we mentioned above, there are gates changing the phase with different values than π. One of these gates is the T gate (Fig.4b). The gate changes the phase about $\pi/4$. When we visualize it in the block sphere we see that the gate is the rotation about the Z-axis as well as the Z gate. The ibmqx2 offers the transposed conjugate of the T gate - T'. T' changes the phase about $-\pi/4$.

A S gate (Fig.4a) is the next gate changing the phase. The gate changes the phase about $\pi/2$. The ibmqx2 also offers the transposed conjugate version of the gate as well as in the T case. The S' moves the phase about $-\pi/2$.

Until now, we've described unary gates. To make complex functionality we need to use some binary gates. The CNOT gate (Fig.5a) is a binary gate which is used to make complex algorithms. The CNOT stands for a control-NOT operation in the standard computer analogy. As Fig.5a shows that the first line is a control line and the second line is called a target line. The control line manages the state of the target line. The result

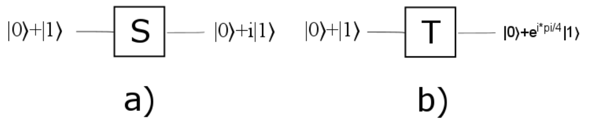


Fig. 4. a) quantum gate S, b) quantum gate T

TABLE I
THE TRUTH TABLE OF CNOT

In		Out	
Control	**Target**	**Control**	**Target**
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

value depends on the value of the control line. The truth table of CNOT is shown in Tab.1

When our quantum program executes all operations what we wanted it is time to get a result. This operation is processed by a special gate. This gate is a M gate - a measurement gate. This gate performs the measurement of the qubit state. The quantum measurement destroys all quantum states and entanglements what we did in the previous steps. After this step we manipulate with the bit, not the qubit. The M gate returns either 0 or 1 with the particular probability of occurrence.

D. Quantum vs Classical Parallelism

Classical computers can't compute some algorithms in the polynomial time. That means huge parallel systems have problems to solve some issues. When we add new computer units to this system the performance of the system increases slowly. After a certain time these additions are not effective. Hash functions or some cryptography methods use this fact. Because the principle of the quantum computer is completely different, QCs can solve these issues.

As explained in the works of Nielsen [5], Heckey [8], Rieffel [9] and others, the classical computers compute a single outcome for input states, but quantum computers compute all possible outcomes for input states. That means, it processes 2^N operations simultaneously, N is the number of qubites.

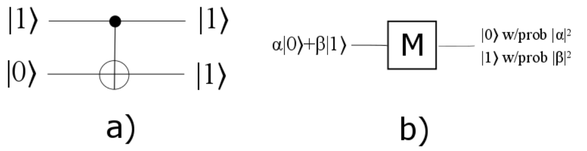


Fig. 5. a) C-NOT, b) Measurement

E. Quantum Decoherence

Schlosshauer [10], Bittner [11] and others [12] explain that a quantum decoherence is a loss of information. This phenomena is due to the environment. The decoherence directly influences results. When we work with the qubit in our application the qubit may change the quantum state "spontaneously" after a certain time. The decoherence stays behind this change, of course. A coherence time is therefore very important for a correct program execution. IBM [12] declares the coherence time at value $1 * 10^{-4}$ seconds. Researches are still trying to improve this time. They want to decrease the quantum decoherence this way and allow the execution of more complex programs at the same time.

F. Error-correction in the quantum program

Classical computers generate error outcomes sometimes. They therefore have mechanisms to detect and repair bad results. Quantum computers also can generate bad outcomes, but the nature of the bad outcomes may be different. Calderbank [13] and Gottesman [14] explain that the errors can be caused by inaccuracy or decoherence. They describe different strategies to reduce errors in the outcomes, such as stabilizer measurements or a bit-flip code method.

III. OPENQASM

An OpenQASM is the way how we can create the quantum program. At the beginning it is important to say the ibmqx2, and a new version ibmqx3 as well, don't implement all operations and statements defined in the OpenQASM. The statement if is not implemented for instance. Cycles are not defined either. Some language constructions which we are used to using in classical computer languages are very challenging to implement at QCs. Cross's work [15] describes the fundamental operations, statements and a general usage of the OpenQASM. Tab.2 shows commands available in the ibmqx2.

A variable q, in the Tab.2, is used as an input argument of operations which we want to process on the quantum computer. c variable is the input argument of operations as well as q. The main difference between them is that c represents the standard bit and q is the qubit. We therefore just use c in two cases. The first case is the initialization of variable. The second is the measurement of the quantum state. To measure the state of the qubit we have to define a source qubit and a destination classical bit.

Operations $u1, u2$ and $u3$ are different. Except for the destination qubit they contain more arguments, such as a pre-phase rotation or a post-phase rotation. All arguments are input arguments. These arguments stand for varying phase rotations. λ is the post-phase rotation, ϕ represents the pre-phase rotation and θ is the Y-rotation. The value range starts at 0 and ends at π.

Gates $u1, u2, u3$ are the physical gates. That means gates, such as Pauli's X gates or Hadamard gates, use these physical gates to realize their functionality. We can denote these non-physical gates as abstract gates. When we use the abstract gate

TABLE II
AN INSTRUCTION SET AVAILABLE IN IBMQX2

Instruction	The meaning	Usage
include	it includes a source file	include ”qelib1.inc”;
qreg	inicialization of qubits	qreg q;
creg	inicialization of bits	creg c;
x	the Pauli’s X gate	x q;
y	the Pauli’s Y gate	y q;
z	the Pauli’s Z gate	z q;
h	the Hadamard gate	h q;
s	the phase gate S	s q;
sdg	the transposed conjugate of S gate	sdg q;
t	The phase gate T	t q;
tdg	the transposed conjugate of T gate	tdg q;
cx	The Controlled-NOT gate	cx q0,q1;
id	The gate preforms an idle operation	id
u1	The first physical gate	u1(λ) q
u2	The second physical gate	u2(ϕ,λ) q
u3	The third physical gate	u3(θ,ϕ,λ) q
barrier	It prevents transformations across a source line.	barrier q,q1,q2;
measure	it measures the qubit value	measure q -> c;

TABLE III
AN INSTRUCTION SET WHICH IS AVAILABLE IN THE IBMQX2

Abstract gate name	Physical gate representation
x	u3(pi,0,pi)
y	u3(pi,pi/2,pi/2)
z	u1(pi)
s	u1(pi/2)
sdg	u1(-pi/2)
t	u1(pi/4)
tdg	u1(-pi/4)
h	u2(0,pi)

it is subsequently transformed to a particular physical gate. The way the gates are transformed to physical gates depends on a particular quantum computer architecture. Table 3. shows the way how the abstract gates are transformed to the physical gates in the ibmqx2.

IV. EXPERIMENTS

We want to show some experiments based on quantum theories in this section.

A. Testing of EPR circle

As we mentioned above the EPR experiment is used to create the quantum entanglement. To verify existence or attributes of the entanglement the Bell experiment [16] is used. The work describes general principles and consequences of the Bell experiment. IBM’s work [17] presents the particular implementation of these experiment on the ibmxq2. The EPR circle consists of two gates. The Fig.6 shows a scheme of the circle. Algorithm 1 demonstrates the implementation in the OpenQASM. As we see in the Fig.6, the first step is the creation of the superposition at the control line of the CNOT. We then apply the CNOT gate. The Hadamard gate ensures a change of control line values. The target line is initialized at the $|0\rangle$. Based on the CNOT truth table we can predict the potential outcomes. We expect either $|00\rangle$ or $|11\rangle$ with similar probability of the occurrence in yhe results. We used 1000 shots in all experiments. The shots mean how many times we want to repeat our algorithm. Let’s look at the Fig.7 where we visualize the results. We see that the result is a little bit unexpected. There are outcomes which we wouldn’t expect. We didn’t apply any error-correction method. This is why we got these unexpected results.

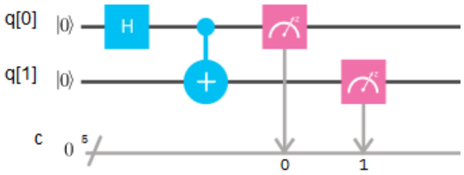


Fig. 6. The scheme of the ERP circle

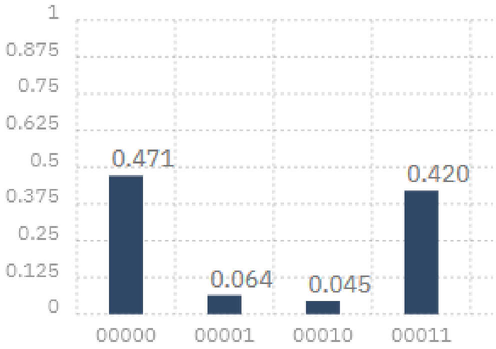


Fig. 7. Results of the ERP experiment

Algorithm 1: Code of the EPR circle

```
include ”qelib1.inc”;
qreg q[2];
creg c[2];
h q[0];
cx q[0],q[1];
measure q[0] -> c[0];
measure q[1] -> c[1];
```

B. Swap gate

The second experiment presents the way how it is possible to perform a swap of two quantum values. Wilmott’s work [18] and Garcia-Escartin’s work [19] show the theoretical concept of the operation. Their approach uses 3 CNOT gates to swap quantum values. The Fig.8 shows a scheme of their approach. The second CNOT is a problem. The ibmqx2 doesn’t allow to use the CNOT in reverse direction as it is shown in the Fig.8. The physical construction of the quantum computer is the reason. IBM declares [20] that there is a way how we can construct a reverse CNOT. Their solution of the reverse

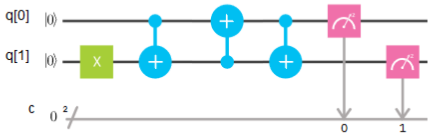


Fig. 8. Scheme of the swap gate

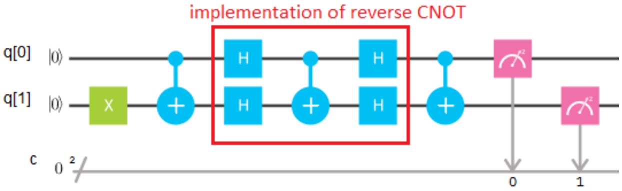


Fig. 9. Scheme of the swap gate implementation in the ibmqx2

CNOT uses 4 Hadamard gates and 1 CNOT. One of things which the Fig.9 shows is the implementation of the reverse CNOT. The reverse CNOT is in the red frame. In general, the Fig.9 presents a scheme of the SWAP operation. Algorithm 2 is a OpenQASM code of this operation. An input of the algorithm is $|01\rangle$ and an outcome is $|10\rangle$. Line 4 ensures that the value of $q[1]$ is changed from $|0\rangle$ to $|1\rangle$. If we want to use the value $|10\rangle$ as the input value we have to change the index of q from 1 to 0 at the line 4. Fig.10 visualizes the result of swap gate experiment. The state $|10\rangle$ has reached the highest value as we expected. Other values are due to we didn't implement any error correction method.

Algorithm 2: Code of the swap gate

```
1  include ”qelib1.inc”;
2  qreg q[2];
3  creg c[2];
4  x q[1];
5  cx q[0],q[1];
6  h q[0];
7  h q[1];
8  cx q[0],q[1];
9  h q[0];
10 h q[1];
11 cx q[0],q[1];
12 measure q[0] -> c[0];
13 measure q[1] -> c[1];
```

C. Experiments with the U3 gate

The U3 gate is one of three physical gates in the ibmqx2. The U3 gate allows us to change the probability of the particular state occurrence in the result. To manage the probability the θ argument is used. The value range of the argument is from 0 to π. When we set up the θ to π*0.9 the probability of value 1 is 90% in the result. Table 4. shows experimental results with varying values of the θ (data are the result of simulations).

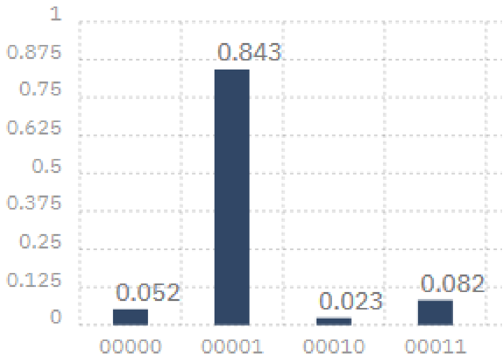


Fig. 10. Result of the swap gate experiment

TABLE IV
RESULTS OF THE U3 GATE EXPERIMENT

θ	1	0
$\theta = \pi * 0.9$	0.869	0.131
$\theta = \pi * 0.8$	0.819	0.181
$\theta = \pi * 0.7$	0.735	0.265
$\theta = \pi * 0.6$	0.589	0.411
$\theta = \pi * 0.5$	0.451	0.549
$\theta = \pi * 0.4$	0.316	0.684
$\theta = \pi * 0.3$	0.206	0.794
$\theta = \pi * 0.2$	0.093	0.907
$\theta = \pi * 0.1$	0.032	0.968

Fig. 11 visualizes these value to a chart. We can see there that measured results are very close to expected values. The code of this experiment is shown in the Algorithm 3. The change of the probability value is an important attribute. Some quantum algorithms can use these attributes, such as quantum neural networks [21], information coding or games [22], [23]. We assume that we can use this attribute to encode complex information to the qubit by this way. Grammar approaches may be useful for this aim. For example: Table 4. shows 9 different states of the qubit. Based on these states we are able to define an alphabet with 9 letters. Works of Moore [24] and Bertoni [25] describe quantum automata and quantum grammars in more detail.

Algorithm 3: Code of the U3 gate experiment

```
include ”qelib1.inc”;
qreg q[5];
creg c[5];
u3(pi*0.1,0,0) q[0];
measure q[0] -> c[0];
```

V. CONCLUSION

This paper is focused on elementary principles of the quantum programming. Although, the present available quantum computer doesn't have a huge performance, it shows fundamental attributes what we can expect in further quantum computer versions as well.

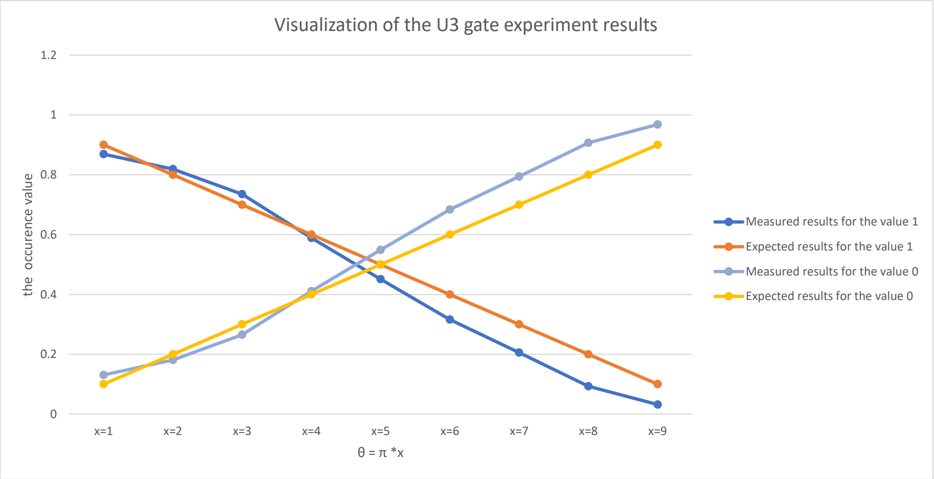


Fig. 11. Visualization of the U3 gate experiment results

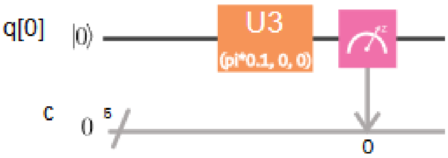


Fig. 12. Scheme of the U3 gate experiment

We performed three experiments where we focused on elementary quantum attributes. We also explained the quantum properties on the experiments. The quantum properties show us the potential of the QCs for processing a huge amount of data.

Acknowledgment

This work was supported by project KEGA 047TUKE4/2016 - Integrating software processes into the teaching of programming.

References

[1] B. Valiron, “Quantum computation: a tutorial,” *New Generation Computing*, vol. 30, no. 4, pp. 271–296, 2012.

[2] R. S. Smith, M. J. Curtis, and W. J. Zeng, “A practical quantum instruction set architecture,” *arXiv preprint arXiv:1608.03355*, 2016.

[3] D. Dong and I. R. Petersen, “Quantum control theory and applications: a survey,” *IET Control Theory & Applications*, vol. 4, no. 12, pp. 2651–2671, 2010.

[4] Y. Patel, “Communication and control for quantum circuits,” 2010.

[5] M. A. Nielsen and I. L. Chuang, “Quantum computation and quantum information,” *Quantum*, vol. 546, p. 1231, 2000.

[6] A. Einstein, B. Podolsky, and N. Rosen, “Can quantum-mechanical description of physical reality be considered complete?” *Physical review*, vol. 47, no. 10, p. 777, 1935.

[7] IBM. (2017) Quantum computing - ibm q. [Online]. Available: http://research.ibm.com/ibm-q/

[8] J. Heckey, S. Patil, A. JavadiAbhari, A. Holmes, D. Kudrow, K. R. Brown, D. Franklin, F. T. Chong, and M. Martonosi, “Compiler management of communication and parallelism for quantum computation,” in *ACM SIGARCH Computer Architecture News*, vol. 43, no. 1. ACM, 2015, pp. 445–456.

[9] E. Rieffel and W. Polak, “An introduction to quantum computing for non-physicists,” *ACM Computing surveys*, vol. 32, no. 3, pp. 300–335, 2000.

[10] M. Schlosshauer, “Decoherence, the measurement problem, and interpretations of quantum mechanics,” *Reviews of Modern physics*, vol. 76, no. 4, p. 1267, 2005.

[11] E. R. Bittner and P. J. Rossky, “Quantum decoherence in mixed quantum-classical systems: Nonadiabatic processes,” *The Journal of chemical physics*, vol. 103, no. 18, pp. 8130–8143, 1995.

[12] IBM. (2017) Entanglement and bell tests. [Online]. Available: https://quantumexperience.ng.bluemix.net/qx/tutorial? sectionId=71972f437b08e12d1f465a8857f4514c&pageIndex=5

[13] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Physical Review A*, vol. 54, no. 2, p. 1098, 1996.

[14] D. Gottesman, “Stabilizer codes and quantum error correction,” *arXiv preprint quant-ph/9705052*, 1997.

[15] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, “Open quantum assembly language,” *arXiv preprint arXiv:1707.03429*, 2017.

[16] J.-W. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger, “Experimental entanglement swapping: entangling photons that never interacted,” *Physical Review Letters*, vol. 80, no. 18, p. 3891, 1998.

[17] IBM. (2017) Entanglement and bell tests. [Online]. Available: https://quantumexperience.ng.bluemix.net/qx/tutorial? sectionId=050edf961d485bfcd9962933ea09062b&pageIndex=3

[18] C. M. Wilmott and P. R. Wild, “On a generalized quantum swap gate,” *International Journal of Quantum Information*, vol. 10, no. 03, p. 1250034, 2012.

[19] J. C. Garcia-Escartin and P. Chamorro-Posada, “A swap gate for qudits,” *Quantum information processing*, vol. 12, no. 12, pp. 3625–3631, 2013.

[20] IBM. (2017) Entanglement and bell tests. [Online]. Available: https://quantumexperience.ng.bluemix.net/qx/tutorial? sectionId=8443c4f713521c10b1a56a533958286b&pageIndex=1

[21] S. Gupta and R. Zia, “Quantum neural networks,” *Journal of Computer and System Sciences*, vol. 63, no. 3, pp. 355–383, 2001.

[22] J. Eisert, M. Wilkens, and M. Lewenstein, “Quantum games and quantum strategies,” *Physical Review Letters*, vol. 83, no. 15, p. 3077, 1999.

[23] H. Guo, J. Zhang, and G. J. Koehler, “A survey of quantum games,” *Decision Support Systems*, vol. 46, no. 1, pp. 318–332, 2008.

[24] C. Moore and J. P. Crutchfield, “Quantum automata and quantum grammars,” *Theoretical Computer Science*, vol. 237, no. 1, pp. 275–306, 2000.

[25] A. Bertoni and M. Carpentieri, “Regular languages accepted by quantum automata,” *Information and Computation*, vol. 165, no. 2, pp. 174–182, 2001.

Software Support for Course in Semantics of Programming Languages

William Steingartner, Mohamed Ali M. Eldojali
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
Email: william.steingartner@tuke.sk
eldojalimohamed@gmail.com

Davorka Radaković
Faculty of Sciences
University of Novi Sad
Novi Sad, Serbia
Email: davorkar@dmi.uns.ac.rs

Jiří Dostál
Faculty of Education
Palacký University Olomouc
Olomouc, Czechia
Email: j.dostal@upol.cz

***Abstract*—Nowadays, computer science increasingly uses formal methods to enhance understanding of complex software systems and to reason about their behavior with respect to a formal specification. To let future generations of software developers and engineers profit from these exciting developments, however, it is necessary to adequately educate and train them in the basics of formal logic and formal language semantics. However, preciously few software tools do exist that substantially aid this educational process. Because the semantics is an integral part of a formal definition of a programming language, we have prepared a package of modules, that help us and to students to understand the most popular semantic method - structural operational semantics. The first module translates a program written in a programming language to abstract machine code, the second module makes reverse translation from code to program source text and the third one emulates stepwise execution of abstract machine code. Our package can be easily extended for other semantic methods.**

***Index Terms*—abstract machine, compiler, learning software, structural operational semantics, transition relation**

I. Introduction

An integral part of formal definition of a programming language is its semantics. Semantics provides a meaning of a program and can help during its design and implementation [1]. There several semantic methods are known, e.g. denotational semantics, operational semantics, action semantics, axiomatic semantics, categorical semantics and others. Each of these methods serves for various purposes, e.g. denotational semantics is useful for language design [2], axiomatic semantics is used for verification purposes, etc. Some of these methods need deeper knowledge of mathematics (denotational semantics) and/or logic (axiomatic semantics) [3]. In this paper we concern with operational semantics, a very popular semantic method that provides meaning of the execution steps of a program and it is better understandable for those who are not familiar with mathematics.

Operational semantics was introduced in [4] and is widely used also by practical programmers. This method defines a meaning of every step of program execution by transition relations between states before and after a given statement execution. In contrast with other semantic methods we have positive experience in teaching this method, because it is understandable for students and software engineers as it requires minimal knowledge of mathematics. Operational semantics incorporates abstract implementation on an abstract machine that enables partial verification of programs [5], [6]. An important tool for explanation this method and for easy understanding by students seems to be a package of programs illustrating translation from program text to the code of abstract machine and vice versa together with step by step execution.

The aim of our paper follows from the ideas mentioned above. We present our package consisting of three modules. Each of modules behaves as separated program and can be run independently. The first module translates a program text written in the simple language $\mathscr{J}ane$ to a code (sequence of instructions) of abstract machine, the second one performs the reverse process, i.e. it provides a program text in $\mathscr{J}ane$ from a code of abstract machine and the third one is an emulator of the stepwise execution of a code on abstract machine. These three programs afforce in significant measure the appreciation of this method by students and help them to achieve skills in defining operational semantics of programming languages.

We start with a short definition of operational semantics in Section II where we present the basic concepts and notation used in this paper. Section III contains the definition of an abstract machine, i.e. the syntax and semantics of the instructions. We define translation functions from the statements of $\mathscr{J}ane$ to the sequences of instructions. Sections IV, V and VI concern with the design and implementation of three programs: from program text to code, from code to program text and stepwise execution of code. This integrated package has intermediate output that is intended to the extension for other semantic methods.

II. Structural operational semantics of $\mathscr{J}ane$

We use in our approach simple imperative language $\mathscr{J}ane$. This language contains arithmetic and Boolean expressions and statements which belong to syntactic domains **Expr**, **BExpr** and **Statm**, resp. In this paper we concern only with statements that have the following syntax:

$$\begin{aligned} S ::= \;& x := e \mid \texttt{skip} \mid S;S \mid \texttt{if}\ b\ \texttt{then}\ S\ \texttt{else}\ S \mid \\ & \texttt{while}\ b\ \texttt{do}\ S \end{aligned}$$

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

where $S \in \mathbf{Statm}$ stands for a statement.

The meaning of program is expressed as a change of state. Therefore the concept of state is introduced. A state is represented as a function from variables to values [7]. The semantic domain of states is

$$\mathbf{State} = \mathbf{Var} \to \mathbb{Z},$$

where $\mathbb{Z}$ is the set of integers. A state is a basic notion of structural operational semantics and it can be considered as an abstraction of computer memory [8].

A value of the state s is only taken (not modified) when a value of arithmetic or Boolean expression is evaluated. The meaning of expressions is defined as follows. For arithmetic expressions the semantic function $\mathscr{E}$ is defined:

$$\mathscr{E} : \mathbf{Expr} \to (\mathbf{State} \to \mathbb{Z});$$

and for Boolean expression the semantic function $\mathscr{B}$ is defined:

$$\mathscr{B} : \mathbf{BExpr} \to (\mathbf{State} \to \mathbb{B}),$$

where $\mathbb{B} = \{\mathbf{tt}, \mathbf{ff}\}$ is the set of Boolean values.

An effect of structural operational semantics comes in light when we define it for statements. The semantics of statements is defined by rules which are transition relations [9]. They describe how a state is changed by execution of a given statement. That means a change of initial state to a final one is a meaning of a whole program.

Transition relations describe individual steps of the statement execution. The transition relation has a form

$$\langle S, s \rangle \Rightarrow \alpha,$$

where α stands either for another configuration $\langle S', s' \rangle$ or for the resulting state s'. The possible outcomes are:

- if the computation of S from s has terminated and the final state is s':
$$\langle S, s \rangle \Rightarrow s';$$
- if the computation of S from a state s is not completed and the execution continues with the substatement S' in a state s':
$$\langle S, s \rangle \Rightarrow \langle S', s' \rangle;$$
- if no rule is applicable, then the execution is stopped and this configuration is a stuck.

The definition of the transition relation $\Rightarrow$ is given by the following axioms and rules:

$$\langle x := e, s \rangle \Rightarrow s[x \mapsto \mathscr{E}[\![e]\!]s] \qquad (1_{os})$$

$$\langle \texttt{skip}, s \rangle \Rightarrow s \qquad (2_{os})$$

$$\frac{\langle S_1, s \rangle \Rightarrow \langle S_1', s' \rangle}{\langle S_1; S_2, s \rangle \Rightarrow \langle S_1'; S_2, s' \rangle} \qquad (3^1_{os})$$

$$\frac{\langle S_1, s \rangle \Rightarrow s'}{\langle S_1; S_2, s \rangle \Rightarrow \langle S_2, s' \rangle} \qquad (3^2_{os})$$

$$\frac{\mathscr{B}[\![b]\!]s = \mathbf{tt}}{\langle \texttt{if}\ \ b\ \ \texttt{then}\ \ S_1\ \ \texttt{else}\ \ S_2, s \rangle \Rightarrow \langle S_1, s \rangle} \qquad (4^{\mathbf{tt}}_{os})$$

$$\frac{\mathscr{B}[\![b]\!]s = \mathbf{ff}}{\langle \texttt{if}\ \ b\ \ \texttt{then}\ \ S_1\ \ \texttt{else}\ \ S_2, s \rangle \Rightarrow \langle S_2, s \rangle} \qquad (4^{\mathbf{ff}}_{os})$$

$$\begin{aligned}&\langle \texttt{while}\ \ b\ \ \texttt{do}\ \ S, s \rangle \Rightarrow\\ &\quad \langle \texttt{if}\ \ b\ \ \texttt{then}\ \ (S; \texttt{while}\ \ b\ \ \texttt{do}\ \ S)\\ &\qquad \texttt{else}\ \ \texttt{skip}, s \rangle\end{aligned} \qquad (5_{os})$$

Given a statement S and a state s, it is always possible to find at least derivation sequence that starts in the configuration $\langle S, s \rangle$: simply applying axioms and rules forever or until a terminal or stuck configuration is reached [7].

Example 1. Consider the statement

$$\texttt{if}\ (x <= y)\ \texttt{then}\ max := y\ \texttt{else}\ max := x$$

and the initial state $s = [x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$. The derivation sequence is as follows:

$$\begin{aligned}&\langle \texttt{if}\ (x <= y)\ \texttt{then}\ max := y\ \texttt{else}\ max := x,\\ &\quad [x \mapsto \mathbf{9}, y \mapsto \mathbf{7}] \rangle \Rightarrow\\ &\langle max := x, [x \mapsto \mathbf{9}, y \mapsto \mathbf{7}] \rangle \Rightarrow\\ &[max \mapsto \mathbf{9}, x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]\,.\end{aligned}$$

It follows from the resulting state, that the maximum of the given input values is $\mathbf{9}$.

□

III. Abstract Machine

A formal specification of a language is necessary for its implementation. And then the question of correctness of its abstract implementation becomes, i.e. the results of AM execution and operational semantics are the same. A usual way in practice is: to translate language into a structured form of assembler code for an abstract machine and then to prove that the abstract implementation is correct. So the basic idea is:

- first, to define the meaning of the abstract machine instructions by an operational semantics;
- second, to define translation functions for all well-defined syntactic constructs in language that will map expressions

and statements in the target language into sequences of such instructions.

The correctness results will then state if we:

- translate a program into code and
- execute the code on the defined abstract machine,

then we get the same results as were specified by the semantic functions of operational semantics [10], [11].

A. The definition of abstract machine

To specify an abstract machine (AM) it is convenient first to present its configuration and next its instructions and their meanings. We follow the obvious definition of abstract machine in [8]. The configuration is defined as a triple containing:

- the sequence of instructions to be executed (in the next step), denoted by c;
- the evaluation stack (for transient data), denoted e; and
- the storage, represented as state function, $s \in \mathbf{State}$ and it is used to hold the values of variables.

Then the configuration is expressed as a triple

$$\langle c, e, s\rangle .$$

An evaluation stack is used to store intermediate values in evaluating arithmetic and Boolean expressions and it is formally defined as a list of values, where the elements come from the domain **Stack**,

$$e \in \mathbf{Stack}$$

which is defined as a transitive closure of the union $(\mathbb{Z} \cup \mathbb{B})^*$.

The syntactic domain of sequences of instructions is denoted as **Code**. The syntax of code $c \in \mathbf{Code}$ is

$$c ::= \varepsilon \mid \mathit{inst} : c$$

The instructions of AM are given by the abstract syntax listed in the following production rule

$$\begin{array}{rl}
\mathit{inst} ::= & \mathtt{PUSH}-n \mid \mathtt{ADD} \mid \mathtt{SUB} \mid \mathtt{MULT} \\
& \mathtt{TRUE} \mid \mathtt{FALSE} \mid \mathtt{EQ} \mid \mathtt{LE} \mid \mathtt{AND} \mid \mathtt{NEG} \\
& \mathtt{FETCH}-x \mid \mathtt{STORE}-x \\
& \mathtt{EMPTYOP} \mid \mathtt{BRANCH}(c,c) \mid \mathtt{LOOP}(c,c)
\end{array}$$

where c is a metavariable ranging over **Code**.

Therefore a configuration is an element of cartesian product

$$\langle c, e, s\rangle \in \mathbf{Code} \times \mathbf{Stack} \times \mathbf{State}.$$

If a program terminates, the final configuration is

$$\langle \varepsilon, \varepsilon, s'\rangle .$$

where ε represents the empty sequence.

The semantics of the instructions of AM is given by an operational semantics and formally specified by a transition system. The transition relation, denoted $\triangleright$, specifies how to execute instructions (for one step):

$$\langle c, e, s\rangle \triangleright \langle c', e', s'\rangle . \qquad (1)$$

The basic idea is, that one step of execution on AM transforms the input configuration into the output one. Formally, the semantics of AM instructions is defined as follows:

$$\begin{array}{ll}
\langle \mathtt{PUSH}-n : c, e, s\rangle & \triangleright \langle c, \mathscr{N}[\![n]\!] : e, s\rangle \\
\langle \mathtt{ADD} : c, v_1 : v_2 : e, s\rangle & \triangleright \langle c, (v_1 \oplus v_2) : e, s\rangle \\
\langle \mathtt{SUB} : c, v_1 : v_2 : e, s\rangle & \triangleright \langle c, (v_1 \ominus v_2) : e, s\rangle \\
\langle \mathtt{MULT} : c, v_1 : v_2 : e, s\rangle & \triangleright \langle c, (v_1 \otimes v_2) : e, s\rangle \\
\langle \mathtt{TRUE} : c, e, s\rangle & \triangleright \langle c, \mathbf{tt} : e, s\rangle \\
\langle \mathtt{FALSE} : c, e, s\rangle & \triangleright \langle c, \mathbf{ff} : e, s\rangle \\
\langle \mathtt{EQ} : c, v_1 : v_2 : e, s\rangle & \triangleright \langle c, (v_1 = v_2) : e, s\rangle \\
\langle \mathtt{LE} : c, v_1 : v_2 : e, s\rangle & \triangleright \langle c, (v_1 \leq v_2) : e, s\rangle \\
\langle \mathtt{AND} : c, t_1 : t_2 : e, s\rangle & \triangleright \begin{cases} \langle c, \mathbf{tt} : e, s\rangle , \\ \quad \text{if } t_1 = \mathtt{tt} \\ \quad \text{and } t_2 = \mathtt{tt} \\ \langle c, \mathbf{ff} : e, s\rangle , \\ \quad \text{if } t_1 = \mathtt{ff} \\ \quad \text{or } t_2 = \mathtt{ff} \end{cases} \\
\langle \mathtt{NEG} : c, t : e, s\rangle & \triangleright \begin{cases} \langle c, \mathbf{ff} : e, s\rangle , \\ \quad \text{if } t = \mathtt{tt} \\ \langle c, \mathbf{tt} : e, s\rangle , \\ \quad \text{if } t = \mathtt{ff} \end{cases} \\
\langle \mathtt{FETCH}-x : c, e, s\rangle & \triangleright \langle c, (sx) : e, s\rangle \\
\langle \mathtt{STORE}-x : c, v : e, s\rangle & \triangleright \langle c, e, s\,[x \mapsto v]\rangle \\
\langle \mathtt{EMPTYOP} : c, e, s\rangle & \triangleright \langle c, e, s\rangle \\
\langle \mathtt{BRANCH}(c_1, c_2) : c, t : e, s\rangle & \triangleright \begin{cases} \langle c_1 : c, e, s\rangle , \\ \quad \text{if } t = \mathtt{tt} \\ \langle c_2 : c, e, s\rangle , \\ \quad \text{if } t = \mathtt{ff} \end{cases}
\end{array}$$

$$\begin{array}{l}
\langle \mathtt{LOOP}(c_1, c_2) : c, e, s\rangle \triangleright \\
\qquad \langle c_1 : \mathtt{BRANCH}(c_2 : \mathtt{LOOP}(c_1, c_2), \mathtt{EMPTYOP}) : c, e, s\rangle
\end{array}$$

where

- $c, c_1, c_2 \in \mathbf{Code}$ are code components,
- $v_1, v_2 \in \mathbb{Z}$ are integer values,
- $t, t_1, t_2 \in \mathbb{B}$ are Boolean values,
- $\mathbf{tt}, \mathbf{ff}$ are shorthands for semantic values of true and false constants.

In the manner of small-steps semantics, we define a computational sequence for AM. Given a sequence c and a state s, a computational sequence for c and s is either

1) a finite sequence

$$\alpha_0 \triangleright \alpha_1 \triangleright \ldots \alpha_k,$$

consisting of configurations satisfying $\alpha_0 = \langle c_0, \varepsilon, s_0\rangle$, and $\alpha_i \triangleright \alpha_{i+1}$ for $0 \leq i < k$, $k \geq 0$, or

2) an infinite sequence

$$\alpha_0 \triangleright \alpha_1 \triangleright \alpha_2 \triangleright \ldots ,$$

consisting of configurations $\alpha_0 = \langle c_0, \varepsilon, s_0\rangle$ and $\alpha_i \triangleright \alpha_{i+1}$ for $0 \leq i$.

An initial configuration always has an empty evaluation stack. We distinguish two situations: a computational sequence is

- either terminating if it is finite and it provides the terminal configuration,

- or non-terminating, if it is infinite or a stuck.

The meaning of a code c can be defined as a partial function

$$\mathcal{M} : \mathbf{Code} \to (\mathbf{State} \rightharpoonup \mathbf{State}),$$

given as follows:

$$\mathcal{M}[\![c]\!]s = \begin{cases} s', & \mathit{if}\ \langle c, \varepsilon, s\rangle \rhd^* \langle \varepsilon, e, s'\rangle\,; \\ \bot & \text{otherwise.} \end{cases}$$

where $\rhd^*$ denotes finite number of execution steps.

When we have a program written in $\mathscr{J}ane$, we need to translate it into AM code. A code for such defined AM is generated by a translation functions. Translation functions serve for these purposes:

$$\begin{aligned} \mathscr{TE} &: \mathbf{Expr} \to \mathbf{Code}, \\ \mathscr{TB} &: \mathbf{BExpr} \to \mathbf{Code}, \\ \mathscr{TS} &: \mathbf{Statm} \to \mathbf{Code} \end{aligned}$$

where $\mathscr{TE}$ translates arithmetic expressions, $\mathscr{TB}$ translates Boolean expressions and $\mathscr{TS}$ translates statements. These functions we implemented in the first program of our package using the definition of translation functions specified in [7], [12].

Now we have defined all concepts that we used in the design and implementation of our package of three modules:

1) from $\mathscr{J}ane$ to AM code,
2) from AM code to $\mathscr{J}ane$ text,
3) executing of AM code.

The next sections contain the description of these modules.

IV. Compiler from $\mathscr{J}ane$ to AM code

In this section we briefly describe the specification and implementation of the first module - a compiler from $\mathscr{J}ane$ to AM code and algorithms of its primary and secondary functions.

A. The module specification

This program is designed as an application that provides the translation of a program written in $\mathscr{J}ane$ language into the code of AM.

An input is a source text that represents program in $\mathscr{J}ane$. We do not consider variables' declarations so the program consists only of a body. An output is a final code which is either a sequence of AM instructions or XML form. The machine comprises only a program store.

The module consists of the obvious compiler phases. The main function of lexical analysis in this application is tokenization, i.e. dividing the program into valid tokens. In this phase, all unnecessary white characters are eliminated. Also the number of brackets is checked.

The next phase is the syntax analysis. We use the top-down parsing method with error recovery which consists of a set of recursive procedures that gradually examine the syntax in more and more detail. A recursive descent parser contains a possibly recursive procedure for each syntactic construct. Then each statement is checked if it matches the appropriate regular expressions. For each statement of $\mathscr{J}ane$ language a unique regular expression is defined. In case the cycle statement (`while`) or conditional statement (`if`) are found, they are matched recursively with appropriate regular expressions.

After the syntax analysis, a very simple semantic analysis is performed: the type mismatch control in assignments and Boolean conditions in statement constructs.

The compiler generates two kinds of output, based on a user's choice. The first kind of output is a sequence of instructions in AM code. This function of compiler is default. The AM instructions are generated according to the translation rules (see Section III-A). The second kind of output is XML document developed for planned future extension of the teaching software. It would allow to take an XML form of an input program and to use it as an input for the other semantic methods, e.g. a construction of the derivation trees in natural semantics or a graphical representation of program in categorical semantics etc.

B. The implementation of the module

The application is developed in Java programming language. Java is an object-oriented, higher-level language that is platform-independent, designed to work with distributed environments on the Internet, as well as for creating standalone applications. It supports many GUI features that make it easier to use and manipulate the user interface. Moreover, it allows creating modular and reusable code.

The main class *GenerateJPJtoAM_UI* provides communication and interaction with user. The class *InputTokenizer* represents a lexical analysis. The class *RegexPatterns* is used in syntax analysis and it provides the regular expressions for matching the keywords of the $\mathscr{J}ane$ language and expressions. The class *Generator* is used as generator of instructions. Here, particular classes for generating the instructions for each statement are derived from the main generator class. These classes define the following translation rules

- *StoreGenerator* - for the assignment;
- *SkipGenerator* - for the empty statement;
- *IfGenerator* - for the conditional statement;
- *WhileGenerator* - for the logical prefix cycle;
- *BooleanGenerator* - for the Boolean expressions in conditional and cycle statements.

The class *XmlGenerator* is used for generating the XML format of a source program.

The general class diagram is depicted in Figure 1.

Example 2. Consider the program source from Example 1. Our decompiler module translates it to the following AM code:

$$\begin{aligned} &\mathtt{FETCH}-y:\mathtt{FETCH}-x:\mathtt{LE}: \\ &\quad\mathtt{BRANCH}\,(\mathtt{FETCH}-y:\mathtt{STORE}-max, \\ &\quad\ \mathtt{FETCH}-x:\mathtt{STORE}-max) \end{aligned}$$

Graphic user interface showing this example is depicted in Figure 2 and produced XML output is in listing in Figure 3. □

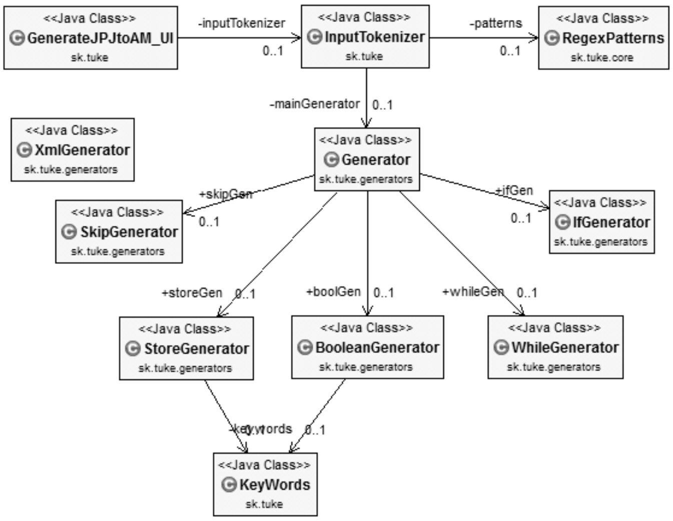
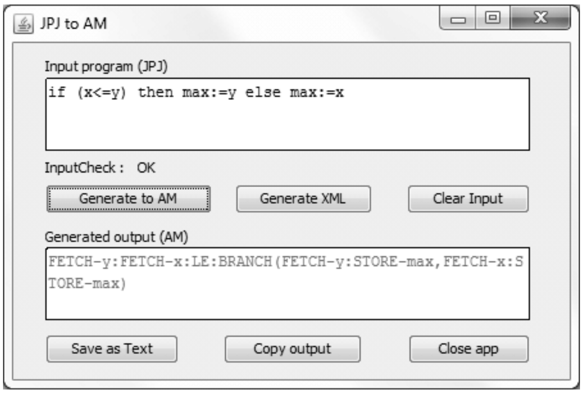


Fig. 1. Class diagram



Input program (JPJ)

```
if (x<=y) then max:=y else max:=x
```

InputCheck : OK

Generated output (AM)

```
FETCH-y:FETCH-x:LE:BRANCH(FETCH-y:STORE-max,FETCH-x:S
TORE-max)
```

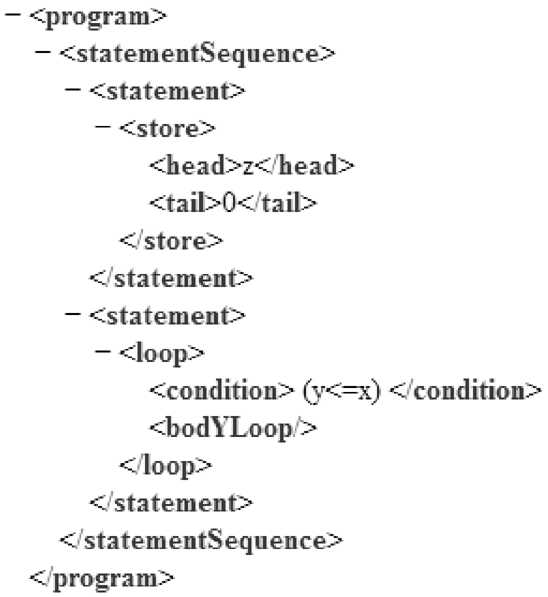
Fig. 2. Graphic user interface

V. Reverse compilation from AM code to *Jane*

The second module - the code generator, prepared mainly for teaching semantics is a reverse compiler from AM code to *Jane* source. This program allows reconstructing code in *Jane* when a sequence of instructions in AM code is given.

The program reads an input and starts with splitting the sequence into particular instructions storing them in a list. The compound instructions BRANCH and LOOP are considered as separate instructions together with their contents, where other recursive splitting to the sequences of instructions are applied and new instruction lists are used.

The next step is recognizing the instructions in the list and building up the *Jane* code. Each instruction is matched with patterns. If an instruction representing an arithmetic or Boolean operation, or instruction representing a value is found, an appropriate symbol or a value is stored in a stack until the end of expression is reached. The correct ending of any well-formed expression is either the instruction STORE (in case of assignment statement), or the instruction BRANCH (in case of conditional statement), or the empty string (in case of expression inside the LOOP instruction). Then the expression is built up from the stack.



```
– <program>
  – <statementSequence>
    – <statement>
      – <store>
          <head>z</head>
          <tail>0</tail>
        </store>
      </statement>
    – <statement>
      – <loop>
          <condition> (y<=x) </condition>
          <bodYLoop/>
        </loop>
      </statement>
    </statementSequence>
  </program>
```

Fig. 3. XML form of input program

If the expression is ended by the instruction STORE, then the arithmetic expression is constructed from stack as right-hand side of the statement, after that the content of the stack is released.

If the expression is ended by the instruction BRANCH, then Boolean expression is constructed from the stack and included into if – then construction, then two other recursive instruction parsings are executed: the first one for then-case, the second one for the else-case.

If the instruction LOOP is found, then again two other recursive code constructions are executed: the first for the Boolean expression, the second one for the body of cycle. Then both strings are included into while – do construction.

If the instruction EMPTYOP is found, the empty statement skip is provided as the output.

An example of reverse compilation and code reconstruction is depicted in Figure 4.

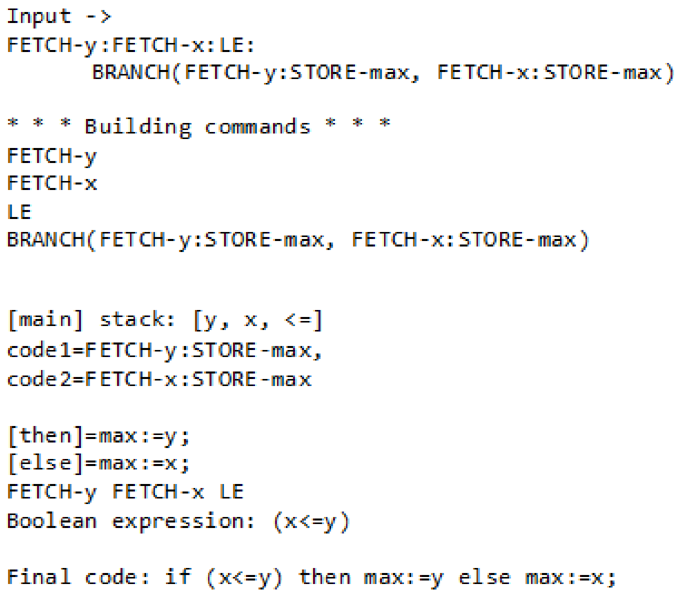
Finally, when the code reconstruction is finished with success, the code in *Jane* can be stored in a text file.

VI. Emulator of AM

The last module in our package is an emulator of AM code. The program is implemented in Java programming language, too.

The program reads the input code in the form of AM instructions. By their processing and splitting into several steps, the output is produced by illustrating the changes performed in an expression stack and states after execution of particular instructions.

At the beginning, an input sequence of instructions is read from the file or is typed manually. After that, the code



```
Input ->
FETCH-y:FETCH-x:LE:
      BRANCH(FETCH-y:STORE-max, FETCH-x:STORE-max)

* * * Building commands * * *
FETCH-y
FETCH-x
LE
BRANCH(FETCH-y:STORE-max, FETCH-x:STORE-max)


[main] stack: [y, x, <=]
code1=FETCH-y:STORE-max,
code2=FETCH-x:STORE-max

[then]=max:=y;
[else]=max:=x;
FETCH-y FETCH-x LE
Boolean expression: (x<=y)

Final code: if (x<=y) then max:=y else max:=x;
```

Fig. 4. Example of code reconstruction

is analyzed by matching the patterns of instructions. Then checking the syntax is easy by using the regular expressions. The sequence is split into separate instructions. Compound instructions BRANCH and LOOP are considered as separate instructions together with their contents - the other sequences of instructions. During this analysis all variables in input source are found and written into table of variables. In this phase, if a syntax error is found, the user is notified about the problematic part of the code. If the syntax checking is completed without errors, then variables can be initialized by user and the analyzed AM source is ready to be executed.

The execution of code provides a complete code processing respecting the semantics of instructions listed in Section III-A. During the stepwise execution the new state on a stack and in memory state are computed. The program displays an actual AM configuration before and after the execution of an actual instruction.

Although the program is syntactically correct, it can contain logical error and during the program execution an infinite cycle can occur. Emulator identifies the number of loops and if this number is greater or equal to the upper limit of using the virtual machine stack, then the cycle is marked as infinite.

At the end of program execution, emulator allows to save the actual state of AM into file, which contains each step of execution together with a state of the stack and memory state.

Example 3. Consider the code in Example 2. This module provides stepwise execution of AM code as it is illustrated in Table I.

Every row in the Table I describes one execution step. The first column contains an instruction to be executed, the second row contains stack and the third contains the state before execution. The last row provides the result state with the empty stack.

Instruction	Stack	State
FETCH $-y$	ε	$[x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$
FETCH $-x$	$\mathbf{7}$	$[x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$
LE	$\mathbf{9} : \mathbf{7}$	$[x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$
BRANCH$(\ldots)$	$\mathbf{ff}$	$[x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$
FETCH $-x$	ε	$[x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$
STORE $-max$	$\mathbf{9}$	$[x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$
ε	ε	$[max \mapsto \mathbf{9}, x \mapsto \mathbf{9}, y \mapsto \mathbf{7}]$

TABLE I
STEPWISE EXECUTION OF AM CODE

□

All three modules were tested on many examples. Because of little space we presented here only a simple one.

VII. Conclusion

We presented in this paper a package of three modules which increase understandability of formal semantic methods, namely structural operational semantics. All the modules are implemented in Java, in object-oriented manner and can be used during lectures, laboratory exercises and also for individual studying. The design of this package enables also future extensions for dealing with other semantic methods thanks to generating XML form. This software can be useful also for practical programmers in the development process of program systems.

Acknowledgments

This work has been supported by Grant No. 002TUKE-4/2017: Innovative didactic methods of education process at university and their importance in increasing education mastership of teachers and development of students' competences.

References

[1] V. Slodičák, V. Novitzká, and M. Lalová, "Algebras on the duality principle for program behavior," *Acta Electrotechnica et Informatica*, vol. 5, no. 3, pp. 50–55, 2005.

[2] D. Schmidt, *Denotational semantics. Methodology for Language Development*. USA: Allyn and Bacon, 1997.

[3] H. Eades, "The semantic analysis of advanced programming languages," Ph.D. dissertation, University of Iowa, 2012.

[4] G. D. Plotkin, "A structural approach to operational semantics," University of Aarhus, Technical Report DAIMI FN-19, 1981.

[5] C. Johansen and O. Owe, "Dynamic structural operational semantics," *CoRR*, vol. abs/1612.00666, 2016.

[6] V. Novitzká, "Logical reasoning about programming of mathematical machines," *Acta Electrotechnica et Informatica*, vol. 8, no. 1, pp. 61–69, 2010.

[7] H. R. Nielson and F. Nielson, *Semantics with Applications. An Appetizer.* Springer Verlag London Ltd., 2007.

[8] D. Turi and G. D. Plotkin, "Towards a mathematical operational semantics," in *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, 1997, pp. 280–291.

[9] G. D. Plotkin, "The origins of structural operational semantics," *The Journal of Logic and Algebraic Programming*, vol. 60-61, pp. pp. 3–15, 2004.

[10] C. B. Jones, "Operational semantics: Concepts and their expression," *Inf. Process. Lett.*, vol. 88, no. 1-2, pp. pp. 27–32, 2003.

[11] P. D. Mosses, "Modular structural operational semantics," 2004.

[12] W. Steingartner and V. Novitzká, *Semantics of programming languages*. Technical university of Košice, 2015, in Slovak.

Event Detection in Hungarian Texts with Dependency and Constituency Parsing and WordNet

Zoltán Subecz
Faculty of Engineering and Computer Science
John von Neumann University
Kecskemét, Hungary
subecz.zoltan@gamf.uni-neumann.hu

***Abstract*— Besides named entity recognition, the detection of events in natural language is an important area of Information Extraction. The detection and analysis of events in natural language texts play an important role in several NLP applications such as summarization and question answering. Most events are denoted by verbs in texts and the verbs usually denote events. But other parts of speech (e.g. noun, participle) can also denote events. In our work we deal with the detection of verbal, infinitival and nominal events. These are the most common events. In this study we introduce a machine learning-based approach with a rich feature set that can automatically detect events in Hungarian texts based on dependency and constituency parsing and WordNet. Additional methods were also applied beside the features, which improved the results and decreased running time. To our best knowledge, ours is the first result for detecting events in Hungarian natural language texts, with dependency and constituency parsing and WordNet. Having evaluated them on test databases, our algorithms achieve competitive results as compared to the current results.**

Keywords*— *Artificial intelligence; Machine learning; Data mining; Text mining; Natural language processing; Information extraction; Event detection; Dependency parsing; Constituency parsing; WordNet

I. Introduction

The detection of events in natural language is an important area of Information Extraction, besides named entity detection [5]. The detection and analysis of events in natural language texts plays an important role in several NLP applications such as summarization and question answering. Finding and analyzing events in a text, the way they relate to each other in time, is crucial to extract the detailed contents of a text.

Most events belong to verbs in texts and the verbs usually denote events. But other parts of speech (e.g. noun, participle) can also denote events. In our work we dealt with the detection of *verbal, infinitival and nominal (nouns and nominalizations) events.* These are the most common events.

In this study we introduce a machine learning-based approach with a rich feature set that can automatically detect events in Hungarian texts with constituency and dependency parsing and WordNet. The model was expanded with rule-based method too. Our system's input is a token-level labeled training corpus. The model's candidates are verbs, infinitives and nouns.

A classifier based upon a rich feature set was applied. Additional methods were used besides the features that improved the results and decreased running time. Our model was tested on five domains of the Szeged Corpus [3].

For English texts *constituency-based* syntactic parsers are usually used for preprocessing, because the English language is strongly configurational, where most of the sentence level syntactical information is expressed by word order. Conversely, Hungarian is a morphologically rich language with free word order. *Dependency parsers* are very useful for morphologically rich languages with free word order, like Hungarian, because they facilitate finding relations between non-neighboring but coherent words. Dependency and constituency parsing was applied for the analysis of our texts. Our hypothesis was that better result can be achieved with dependency parsing than with constituency parsing.

In our system we applied the *Hungarian WordNet* [8] for the semantic characterization of the examined words. Since several meanings may belong to one word form in WordNet, we performed word sense disambiguation (WSD) among the particular senses with the *Lesk algorithm.* [5]

To our best knowledge, ours is the first result for *detection of events in Hungarian* natural language texts, with constituency and dependency parsing and WordNet. Having evaluated them on test databases, our algorithms achieve competitive results as compared to the current results.

II. Related Work

Several papers are concerned with event detection. The most studies focused only on particular events (e.g. business events).

EVITA [9] was one of the first event recognition tools. It recognizes events by combining linguistic and statistical techniques. It uses manually encoded rules based on linguistic information as main features. It also uses WorldNet classes to those rules for nominal event recognition. For sense disambiguation of nouns, it utilizes a Bayesian classifier.

Bethard et al. [1] developed a system for event recognition. They adopted syntactic and semantic features, and formulated the event recognition task as classification in the word-chunking paradigm. They used a rich set of features: textual, morphological, syntactic dependency and some selected

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

WordNet classes. They implemented a Support Vector Machine (SVM) model based on those features.

Llorens et al. [6] presented an evaluation on event recognition. They added semantic roles to features, and built the Conditional Random Field (CRF) model to recognize events.

Marsaic [7] focused only on verbal event detection and classification.

Boguraev et al. [2] presented a machine-learning based approach for automatic event annotation. They set out the task as a classification problem, and used a robust risk minimization (RRM) classifier to solve it. They used lexical and morphological attributes and syntactic chunk types in bi- and tri-gram windows as features.

Jeong et al. [4] used a dependency parser, but investigated the relation only between the candidate noun and the direct verb. They used combined features, applying them for verb + relation type pairs. They used WordNet, but without word sense disambiguation. They applied the MaxEntclassification algorithm with the following feature sets: basic, lexical, semantic, and dependency-based features.

For Hungarian text Subecz [10,11] detected events, but he used only dependency parsing and his methods and results were simpler, not so comprehensively worked-out.

III. Events

Most events belong to verbs in texts and the verbs usually denote events. But other parts of speech (e.g. noun, participle) can also denote events. In our work we dealt with the detection of verbal, infinitival and nominal (nouns and nominalizations) events. These are the most common events.

Examples for verbal and infinitival events: *olvas* (read), *olvasni* (to read), *alszik* (sleep), *aludni* (to sleep). There is ambiguity in relation to events. Not all verbs and infinitives can be considered as event-indicator (e.g. *tud* [know], auxiliaries, e.g. *akar* [want], etc.), thus special attention is needed to filter out them. The word's analysis is not sufficient, so the context of the words should be analyzed too. Other examples: *dob* (verb: throw, noun: drum), *vár* (noun: castle, verb: wait)

Examples for nominal events: *futás* (running), *építés* (building), *írás* (writing), *háború* (war), *ünnepség* (feast). There are words (e.g. *írás* 'writing') that denote events in some sentences, but non-events in others, so the context of the words should be analyzed. Nominal events have two main groups: deverbal and non-deverbal nouns. Deverbal nouns: *írás* (writing), *futás* (running). Non-deverbal nouns: *háború* (war), *ünnepség* (feast). The deverbal nouns have two main types: events and results. These nouns are often ambiguous. These are events in some sentences and results in other sentences. For example the noun *writing* is an event in the following sentence: *The writing began at 5 o'clock.* But it is a result in the following sentence: *We have read the writing*. Because of the ambiguity, the context must be analyzed too.

IV. Environment

One part of the Szeged Corpus [3] was used in our application from the following domains: *business and financial news, fictions and compositions of pupils, computer texts, newspaper articles, legal texts*. For training and evaluating 10 fold cross validation was used. The texts were annotated by two linguists. The inter-annotator agreement was Kappa = 0.87 for verbal and infinitival events and 0.7 for nominal events.

The *Weka*[1] data mining suite were employed for machine learning. For linguistic preprocessing segmentation, morphological analysis, POS-tagging and dependency parsing we applied the *Magyarlanc* 2.0 program package [14]. The Magyarlanc package creates morphological analyses, but the HunMorph package [13] creates more detailed analysis in some cases, so this was also used. Thus two morphological parsers were applied. For nominal event detection we used the Named Entity Recognition tool of the Natural Language Processing Group of the University of Szeged [12].

As we can see at the related works, dependency parsing was used by other studies too. But they examined only the candidate and the direct parent and children in the dependency tree. We examined also the relation between the candidate and the further verbs in the tree. The model's candidates are the verbs, infinitives and nouns.

In our system we applied the Hungarian *WordNet* [8] for the semantic characterization of the examined nouns. The semantic relations of the WordNet hypernym hierarchy were used.

A. Statistical data

Verbs and infinitives: 5,000 sentences, 10,628 verbs and infinitives. The annotators labeled 6,479 of them as event.

Nouns: 10,000 sentences. Number of candidates (nouns): 48,388. Number of positive candidates (event nouns): 7,626. The group of the noun candidates was divided into two main parts on the basis of the similar features. The deverbal nouns got into the first, and the others got into the second group. Deverbal candidates: 5,325, positive deverbal candidates: 4,169. Non-deverbal candidates: 43,063, positive non-deverbal candidates: 3,457.

B. Difference between constituency and dependency parsing

A *constituency parse tree breaks* a text into sub-phrases. Non-terminals in the tree are types of phrases, the terminals are the words in the sentence, and the edges are unlabeled. This parsing is based on the constituency relation. The parse tree is the entire structure, starting from S and ending in each of the leaf nodes.

A *dependency parse* connects words according to their relationships. Each node in the tree represents a word, child nodes are words that are dependent on the parent, and edges are labeled by the relationship. The main verb is the root element. If the candidate consists of several words, then these

[1] http://www.cs.waikato.ac.nz/ml/weka/

words compose a *sub-tree* within the main tree. The sub-tree is attached to the main tree by its *head word*.

V. Classification

We applied *binary classification* for the problem. Decision tree (J48) algorithm of the Weka data mining suite was used. In the first case, the candidates were nouns, in the second, verbs and infinitives. We assigned features for the candidates on the training and evaluation set. Our method applied a rich feature set. The commonly used features of the event detection tasks were also employed. Besides them, our feature set was extended with new ones. (we haven't seen other authors use them in event detection) The *new features* were selected according to the characteristics of the *Hungarian language*. The [NEW] notation was used for new features and methods. The constituency and dependency parsing and the Hungarian Wordnet were also applied for the features.

A. Feature Set

All of the following features were applied for the nominal candidates. Not all of them were applied for the verbal and infinitival candidates because of their irrelevance.

Surface features: *Bigrams and Trigrams*: The character bigrams and trigrams at the end of the examined words. *PositionInSentence*: The sequential number of the candidate in the sentence. *UpperCaseLetterInsideSentence*: True if the candidate is not at the beginning of the sentence and has uppercase letter at start. In this case the nouns are usually named entities, thus this feature can indicate non-eventive nature. *NamedEntity* [NEW]: Whether the word is named entity (NE). (with the NE recognition tool) This can suggest that the candidate is non-event.

Morphological features-1: Since the Hungarian language has rich morphology, several morphology-based features were defined. For this group the morphological parser of the Magyarlanc package were used. We utilized the MSD codes (morphological coding system) of the event candidates for the following morphological features: *type, mood, case, tense, person of possessor, number, definiteness. Lemma*: the candidate's lemma. *Deverbal*: whether the candidate is derived from a verb. *PrevPOS, SubPOS*: the POS codes of the previous and the subsequent words. *NearestVerbLemmaInSentence*: The lemma of the verb nearest to the candidate in the sentence. *Stem*: the deverbal candidate's stem.

Morphological features-2 [NEW: two morphological parsers together]: The HunMorph morphological parser was applied in this group. *DeverbalHunMorph*: whether the noun is derived from a verb. *DeverbalSuffix*: The derivational suffix of a deverbal noun. *StemHunMorph*: the deverbal candidate's stem.

Morphological features-3: The morphological parsers give several parsing possibilities for ambiguous words. The derivational and inflectional suffixes of the candidate were given in this group for both parsers separately.

Dependency tree features-1 [NEW]: These features were extracted from the dependency tree. *EdgeType*: The relationship between the candidate and the word above it in the tree (For example: SUBJ, OBJ, COORD). *EdgeTypeNE*: Whether the label (EdgeType) is named entity (NE). The relationship type is NE between the words of multiword named entities. This can suggest that the candidate is non-event. *LemmaAbove*: The lemma of the parent word of the candidate in the tree. *VerbLemmaAbove*: The lemma of the parent verb of the candidate (if any). *DirectVerbConnection-EdgeType*: The dependency label between the candidate and its parent verb (if any) (For example: SUBJ, OBJ). *NearestVerbLemmaAboveCandidate* and *NearestVerbAbove-CandidateDistance*: The lemma of the nearest verb above the candidate in the tree and the distance between the verb and the candidate. *CandidateSubTreeTokenNumbers*: The number of nodes belonging to the candidate's sub-tree. *Word-AboveCandidateEdgeType:* The dependency label between the parent word of the candidate and its parent word. (For example: TLOCY) A temporal expression belonging to the candidate is usually above the candidate in the tree and its dependency label indicates that it is a temporal expression. The presence of a temporal expression above the candidate can indicate its eventive character.

Dependency tree features-2[NEW]: If there is a verb above the candidate in the dependency tree, but the connection is not direct, the path between the candidate and the verb was characterized in detail. *POSPath*: The parts of speech of the nodes between the candidate and the verb were written one after the other. For example: C↑S↑V↑C↑V↑V. *LemmaPath*: The lemmas between the candidate and the verb were written in a sequence. For example: napoztatás↑és↑törölgetés↑hajszárító↑megszárít. *EdgeTypePath*: The sequence of the dependency labels between the candidate and the verb. For example: OBL↑COORD↑SUBJ↑COORD↑CONJ↑.

Constituency tree features[NEW]:These features were extracted from the constituency tree. *ConstNodeAboveThe-Candidate*: Type of the node above the candidate. ConstNumberOfWords: The number of words of the candidate. Multiword named entities contain several words. *ConstNodePath*: Path of the nodes between the candidate and the nearest verb. (For example: NP↑NP↑V) *ConstGoverning-Category*: Type of the node nearest to the top node on the path between the candidate and the nearest verb.

Bag of words features-1[NEW]: The bag of words (BoW) model was used for the characterization of word groups. *CandidateSubTreeLemmasBoWAverage*: The candidates usually have a sub-tree. This feature characterizes not only the sub-tree's headword but also the other words of the sub-tree. The lemmas of the sub-tree were represented with the bag of words model. First, we calculated for each lemma in the training set the probability of the particular lemma belonging to a positive candidate's sub-tree. Second, for each candidate we calculated the average of these probabilities of the lemmas of the candidate's sub-tree. A high average value indicates that the lemmas of the candidate's sub-tree are important as eventiveness is concerned. *CandidateSubTreeLemmasBoW-Max*: Similar to the previous feature, but at the second step for each candidate we chose the maximum of these probabilities that belong to the lemmas of the candidate's sub-tree. A high maximum probability indicates that there is at least one lemma in the candidate's sub-tree which is highly important as eventiveness is concerned. *LemmasUnderCandidateBoWAverage* and *LemmasUnder-CandidateBoWMax:* Similar to the previous features, but in this case not all of the lemmas of the candidate's sub-tree were investigated, but only those lemmas of the sub-tree that are children of the candidate in the bag of words model. *EdgeType-sUnderCandidateBoWAverage* and *EdgeTypesUnderCandidateBoWMax*: Similar to the previous ones, the dependency labels between the candidate and the lemmas of the candidate's children were represented with bag of words. *LemmaBetweenCandidateVerbBoWAverage* and *LemmaBetweenCandidateVerbBoWMax*: For these features the lemmas between the candidate and the nearest verb above the candidate got into the bag of words. *ConstAbove1LevelSubTreeBoWAverage* and *ConstAbove1LevelSubTreeBoWMax*: Lemmas of the subtree of first level node above the candidate in the constituency tree characterized with a bag-of-words model. *ConstAbove2LevelSubTreeBoWAverage* and *ConstAbove2LevelSubTreeBoWMax*: Lemmas of the subtree of the second level node above the candidate in the constituency tree characterized with a bag-of-words model.

Bag of words features-2[NEW]: For these features the lemmas were represented in the bag of words from the sentence, not from the dependency tree. *WindowNLemmasBoWAverage* and *WindowNLemmasBoWMax*: The lemmas before and after the candidate in N size window in the sentence were characterized with a bag of word model, using N=3 and N=5. *InSentencetoNearestVerbBoWAverage* and *InSen-tencetoNearestVerbBoWAverageMax:* Lemmas between the candidate and the nearest verb characterized with a bag-of-words model.

WordNet features: For these features the semantic relations of the Hungarian WordNet [8] hypernym hierarchy was used. Since several meanings may belong to a word form in the WordNet, we performed word sense disambiguation (WSD) among the particular senses with the Lesk algorithm [5]. Synsets in the WordNet contain definition and illustrative sentences. In the case of polysemous event candidates, we counted how many words from the syntactic environment of the event candidate can be found in the definition and illustrative sentences of the particular WordNet synset (neglecting stop words). The sense containing the highest number of common words was chosen.

WordNet features-1: *UnderEventKinds*: There is an artificial synset in the Hungarian WordNet, which contains mainly events in its hyponym hierarchy. But some of them are not real events and there are events also besides them. This feature indicates whether the given word belongs to the hyponym hierarchy of this synset.

WordNet features-2[NEW]: *WordNetBoWAverage* and *WordNetBoWMax*: Similar to the Bag of words features, in this case the words belonging to the hypernym hierarchy of the candidate were put into a bag of words. *WordNetBoWMaxSynset*: We selected the synset from the hypernym hierarchy of the candidate that belongs with the highest probability to the hypernym hierarchy of events.

WordNet features-3[NEW]: *WordNetHypernymSynsets-TrainingSet* (binary): First the synsets of the hypernym hierarchy of event candidates from the training set were collected to a set. Then for each candidate it was marked whether at least one of the synsets from the hypernym hierarchy belongs to this set.

WordNet features-4[NEW]: First lemmas which were events in the training set with the highest probability rate were collected to a set. Then for each candidate it was indicated whether its lemma is under the hyponym hierarchies of the set's members.

Bag of words features-3[NEW]: First we selected the best members from the bag of words to a separate set for every case of the Bag of words 1-2 groups, i.e. those members that belong to events with the highest rate probability. Then we indicated, with the following features, whether the given bag of words of the given candidate contains at least one member of the particular set. *BestWordNetSynsets*: This feature indicates whether the synsets of the candidate's hypernym hierarchy is included in the BestWordNetSynsets set. *BestSubTreeLemmas*: It indicates if there is at least one lemma among the lemmas of the candidate's sub-tree which is in the BestSubTreeLemmas set. *BestLemmasPathToVerb*: It indicates if there is a lemma among the lemmas between the candidate and the nearest verb in the dependency tree which is in the BestLemmasPathToVerb set. *BestLemmasWindowN*: It indicates if there is a lemma before or after the candidate in a window of N which is in the BestLem-masWindowN set. This was calculated for the cases N=3 and N=5.

List features: *AboveLemmaTemporalList*: We collected frequent temporal expressions to a list (e.g. *előtt* [before], *után* [after], *folyamán* [during]). There are often noun events under these expressions in the dependency tree. This feature indicates if there is such an expression above the candidate. *VerbAboveAspectualList*: We collected frequent aspectual verbs to a list (e.g. *kezd* [begin], *folytat* [continue]). The nouns under such verbs in the dependency tree are often events. This feature indicates if there is such a verb above the candidate.

Combined features–2 members: For these features two previous features were concatenated. *LemmaAbove+EdgeType*: The eventiveness of a word is often more correctly predicted if the lemma above the candidate and the relation between them are together examined, as opposed to the case if these were examined separately. Similarly, the following features were investigated together: *VerbLemmaAbove+EdgeTypeOBJ*, *VerbLemmaAbove+EdgeTypeSUBJ*, *LemmaAbove+BestWordNetSynsets*, *VerbLemmaAbove+BestWordNetSynsets*.

Combined features–3 members: Similar to the previous ones, in this case three features were concatenated. *LemmaAbove+EdgeType+BestWordNetSynsets*, *VerbLemmaAbove+EdgeType+BestWordNetSynsets*.

Combined features–N members: *BestNProbabilityAverage*: For cases N=1, 3, 5, 10, 20, 50, 100: based on the training set we calculated eventive probability for every instance of every feature. Our hypothesis was that the more high probability features a candidate has, the more probable that the candidate is an event. We calculated the highest probability average of N for the given candidate.

B. Rule based method

For the verbal and infinitival candidates we completed our machine learning technique also with a rule based method. There were several expressions in the legal texts where the verb usually indicates event in other contexts, but not in the legal context. For example: *A törvény* ***kimondja****,hogy...* (*The law* ***states*** *that...*) We defined rules for such cases. An example for such a rule: If Subject = "law" And Candidate = "state" Then Candidate ≠ Event.

C. Additional methods [NEW]

All of them were useful based on the results, so these can be useful in other NLP tasks.

Using the probabilities of the base features. The features for the candidates were selected using *two main methods*. *In the first method* we used the base features introduced in the previous section. *In the second method*, instead of the base features, we applied probabilities calculated from the base features from the training set. In the case of each feature instance, we calculated on the basis of the training set how many times it occurred and how many times the candidate was *positive.* Relying on this, the positive-rate was calculated to that feature instance. We did not give the base feature to the classifier, but instead its probability was given. The higher the probability, the more probable that the candidate is an event. With this method *the size of the classifier's vector space* and thus *the running time* were *significantly reduced*, compared to the first method. This is especially useful in the development phase. Comparing the two methods, most of the time *the probability method* gave better results and the running time was *70-80 times faster* than with the base features.

Forming groups from the candidates for nouns. The classifier can find more easily the rules in a dataset where the members are *similar* than in a dataset which contains diverse members. This is why it is worth grouping the candidates into smaller, similar parts (facilitating the work of the classifier) then summarizing the TP, TN, FP, FN results of the groups. Thus, the candidates were grouped according to two main criteria. First *the candidates were grouped into two parts*: deverbal and non-deverbal nouns since the members of these groups behave differently. The group of deverbal nouns contains events in a bigger rate. The second grouping was performed on the basis of the candidates' lemmas. In this case three groups were created. The first group contains lemmas which were usually events in the training set. The second group contains the lemmas of the other candidates from the training set. The third group contains those lemmas of the test set which were not in the training set. Thus 2*3=6 groups were created, and the classification was performed for each group. For each group separately 10 fold cross validation was used for training and evaluating. Then we summarized the TP, TN, FP, FN results of the groups.

Using the lemmas' positive probability for improving the classification result. In the cases when we got weak results (usually due to the low recall), after the classification, we modified the label of candidates which were usually events in the training set to positive.

TABLE I. RESULTS ON THE WHOLE CORPUS (%)

	Precision	Recall	F-measure
Nouns	81.31	68.16	**72.83**
Verbs and infinitives	95.48	95.93	**95.67**

As can be seen from the results, these additional methods improved our results and reduced running time.

VI. RESULTS

For evaluation, the precision (P), recall (R) and F-measure (F) metrics were used.

A. Baseline methods

Baseline solutions were applied for the examination of our model's efficiency.

For verbs and infinitives two baseline solutions were applied. At the first one, every verb and infinitive was treated as event. At the second one, only those verbs and infinitives were treated as event that is not copulas or auxiliary verbs. Our first baseline method achieved an F-measure of 79.45, the second one 84.37.

For nouns as *baseline method* the deverbal candidates were treated as positive, the others as negative. Its result was F-measure: 55.52.

According to the following results, *our machine learning model by far outperformed the Baseline methods.*

B. The results of our model

Our machine learning method achieved the following result on the whole corpus with the given feature set and additional methods with the classifiers. (Table I.)

For nouns without the additional methods we got the following result: precision: 71.43 recall: 61.49, *F-measure*: 65.95. As shown, we could achieve a significant *improvement with the additional methods.* 80% of the improvement was due to the candidates' grouping, while the other part came from correcting the result of classification. When grouping was performed just on according to the first criterion, the model achieved a much better result (F-measure: 85.58) with only deverbal candidates than with only non-deverbal candidates (F-measure: 40.42).

Our model was run separately on the five sub-corpora. These results can be found in Table II. We got the best results on the *newspaper articles* domain, and the worst on the *legal texts*.

TABLE II. RESULTS ON THE SUB-CORPORA (F-MEASURE, %)

Sub-corpus	Nouns	Verbs and infinitives
Fiction and compositions of pupils	76.15	98.66
Newspaper articles	77.35	98.72
Business and financial news	76.23	98.57
Computer texts	73.28	96.32
Legal texts	67.82	88.05

C. Results for Ablation Analysis

We examined the efficiency of the particular *feature groups* with an *ablation analysis* for all of the five cases when the target words were not grouped. In this case the particular feature groups were left out from the whole feature set, and we trained on the basis of the features that had remained. The results can be found in Table III. The figures of the table show how the results changed after leaving out the particular feature group. Negative numbers indicate that the investigated feature group has positive influence on event recognition.

As we can see in Table III, almost all feature groups had positive impact to the model's performance. The *WordNet* and *Dependency tree* features had the best impact, but the *Bag of words* and *Morphological* features also improved a lot.

D. Comparing the results with the related works

For verbs and infinitives Bethard [1] achieved 88.3%, Llorens et al. [6] achieved 91.33% F-measure. For nouns Jeong et al. [4] achieved 71.8% F-measure. Compared with the related works, our result (F-measure = 72.83% for nouns and 95.67% for verbs) is considered good.

Discussion, Conclusions

In this paper, we introduced our machine learning approach based upon a rich feature set, which can detect events in Hungarian texts. We tested our methods on 5 domains of the Szeged Corpus.

TABLE III. RESULTS FOR ABLATION ANALYSIS - CONTRACTION (%)

Left out features	Difference in F-measure
Surface features	-0.25
Morphological features	-1.57
Dependency tree features	-1.75
Constituency tree features	-1.1
WordNet features	-4.8
Bag of words features	-1.62
List features	0.0
Combined features	-1.2

In our feature set, which is based upon a rich feature space, surface, morphological, dependency tree, constituency tree, WordNet, bag of words, list and combined features were used. Beside these feature groups additional methods were applied, which improved our model's efficiency and reduced running time. The best results were achieved on the newspaper articles and the worst on the legal texts. Its reason is that the newspaper articles contain simpler sentences and the legal texts contain more complicated sentences. The WordNet feature had the best impact for the results. Having evaluated our methods on test databases, our algorithms achieve competitive results as compared to the current English and other language results.

Acknowledgment

This research is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at Pallas Athena University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

References

[1] S. Bethard, J. H. Martin. "Identification of Event Mentions and Their Semantic Class." In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 146–154. Association for Computational Linguistics (2006)

[2] B. Boguraev, R. Ando. "Effective Use of TimeBank for TimeML Analysis." In Annotating, Extracting and Reasoning About Time and Events, ed. Frank Schilder, Graham Katz, and James Pustejovsky, 4795:41–58. Springer Berlin / Heidelberg. (2007)

[3] D. Csendes, J. Csirik, T. Gyimóthy: The Szeged Corpus: A POS Tagged and Syntactically Annotated Hungarian Natural Language Corpus in Proc. of the Seventh International Con-ference on Text, Speech and Dialogue (TSD 2004), Brno, Czech Republic 8-11 September, pp. 41-49 (2004)

[4] Y. Jeong, S. Myaeng: Using Syntactic Dependencies and WordNet Classes for Noun Event Recognition. In: The 2nd Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web in Conjunction with the 11th International Semantic Web Conference, pp. 41–50 (2012)

[5] D. Jurafsky and J. H. Martin: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice-Hall, Upper Saddle River, NJ, (2000).

[6] H. Llorens, E. Saquete, B. Navarro-Colorado. "TimeML Events Recognition and Classification: Learning CRF Models with Semantic Roles." In Proceedings of the 23rd International Conference on Computational Linguistics, 725–733. Association for Computational Linguistics. (2010)

[7] Marsic, G.: Temporal processing of news: annotation of temporal expressions, verbal events and temporal relations. PhD thesis, University of Wolverhampton (2011)

[8] M. Miháltz, Cs. Hatvani, J. Kuti, Gy. Szarvas, J. Csirik, G. Prószéky, T. Váradi: Methods and Results of the Hungarian WordNet Project. In A. Tanács, D. Csendes, V. Vincze, C. Fellbaum, P. Vossen, eds.: Proceedings of the Fourth Global WordNet Conference (GWC 2008), Szeged, University of Szeged (2008) 311-320

[9] R. Saurí, R. Knippen, M. Verhagen, J. Pustejovsky. "Evita: a Robust Event Recognizer for QA Systems." In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 700–707. Association for Computational Linguistics. (2005)

[10] Z. Subecz: Detection and Classification of Events in Hungarian Natural Language Texts. Proceedings of the 17th International Conference, TSD 2014, Brno, Czech Republic. Springer Lecture Notes in Computer Science Volume 8655, 2014, pp 68-75

[11] Z. Subecz: Automatic Detection of Nominal Events in Hungarian Texts with Dependency Parsing and WordNet. Proceedings of the 22nd International Conference, ICIST 2016, Druskininkai, Lithuania. Springer Communications in Computer and Information Science Volume 639, 2016, pp 580-592

[12] G. Szarvas, R. Farkas, A. Kocsor: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: The Ninth International Conference on Discovery Science 2006, LNAI 4265

[13] V, Tron. A. Kornai. G. Gyepesi. L. Németh. P. Halácsy. D. Varga: Hunmorph: Open source word analysis. In Proceedings of the Workshop on Software, Software ’05, pp. 77–85, Stroudsburg, PA, USA. Association for Computational Linguistics. (2005)

[14] J. Zsibrita, V. Vincze, R. Farkas: Magyarlanc: A Toolkit for Morphological and Dependency Parsing of Hungarian. (2013) In: Proceedings of RANLP 2013, pp. 763-771.

IDE-Independent Program Comprehension Tools via Source File Overwriting

Matúš Sulír, Jaroslav Porubän and Ondrej Zoričák
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Email: {matus.sulir,jaroslav.poruban}@tuke.sk,
ondrej.zoricak.2@student.tuke.sk

Abstract—**Traditionally, we have two possibilities to design tools for program comprehension and analysis. The first option is to create a standalone program, independent of any source code editor. This way, the act of source code editing is separated from the act of viewing the code analysis results. The second option is to create a plugin for a specific IDE (integrated development environment) – in this case, a separate version must be created for each IDE. We propose an approach where information about source code elements is written directly into source files as annotations or special comments. Before committing to a version control system, the annotations are removed from the source code to avoid code pollution. We briefly evaluate the approach and delineate its limitations.**

I. Introduction

A programming language should serve as a mean of communication between a human and a computer. The valid source code of a program, by definition, contains all required information to be unambiguously compiled and executed on a computer. On the other hand, the source code is often understood by humans with great difficulties, or not at all. This can be attributed to a lack of information useful for humans contained in the code.

To improve program comprehension, a vast variety of tools analyze the source code, a running program, the programmer's interaction, the version control system and other external systems. Then, they associate the obtained metadata with parts of the source code. In our article [1], we performed a systematic mapping study of such approaches and tools. We surveyed existing literature related to labeling (augmenting) the source code and categorized the approaches into a taxonomy with four dimensions: source, target, presentation and persistence.

To clarify the mean of source code labeling, let us illustrate it with specific examples. Stacksplorer [2] statically analyzes the source code and visually annotates method definitions in the IDE with lists of their callers. Senseo [3] displays method callers obtained using dynamic analysis; in addition, it shows metrics like execution frequencies or memory consumption of methods. IDE plugins such as Deep Intellisense [4] and Rationalizer [5] show metadata like an author, time or message of the last commit next to each source code line.

A. Motivation

Currently, when the aim of a third-party tool is to label parts of the source code with metadata, the tool authors have two main implementation possibilities:

- an IDE (integrated development environment) plugin
- or a standalone tool.

Each of these two possibilities has its advantages and shortcomings.

IDE plugins allow for the utilization of a tight integration between the source code, the current IDE features and the new feature of the tool. For example, we can represent a comprehensibility metric with a light or dark red background directly in the editable source code view of the IDE [6]. The background color supplements the existing source code highlighting and the programmer can view this augmentation and edit the source code at the same time, without switching between two separate tools.

However, plugins have also disadvantages. Since individual IDEs have very different plugin development interfaces, large parts of plugins must be developed individually for each supported IDE. For example, to support only some of the most popular Java IDEs, we must create a separate plugin for Eclipse, IntelliJ IDEA, NetBeans and JDeveloper. The situation becomes even worse when we have to consider also text editors like jEdit, Vim and Emacs. Finally, there are slight differences between individual versions of the same IDE, so a plugin developed for Eclipse 4.7 may be incompatible with Eclipse 4.3 and vice versa.

The second possibility is to create a standalone tool which displays the source code and metadata separately, without any connection with an IDE. This poses a cognitive burden on the developer – he must switch back and forth between two tools, navigate to correct parts in both of them and mentally connect the corresponding parts. This is called a split-attention effect [7].

On the other hand, the advantage of separate tools is a lower creation and maintenance cost for the tool author.

B. Synopsis

In this paper, we would like to present an approach which does not require a creation of a separate plugin for each

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

IDE, while at the same time, the metadata will be displayed near related source code parts and directly in the IDE. This can be achieved by tools writing the metadata directly in the source code files. We will limit the metadata only to textual information and write it in a form of Java annotations or source code comments next to the corresponding source code elements or lines. When the programmer will no longer need them, or when the source code will be about to be committed into a version control system, the metadata will be removed from the source code.

In our previous work [8], we described an approach where run-time metadata (obtained using dynamic analysis) were written directly into source code using Java annotations. We distinguished between two operation modes:

- local-only workflow
- and shared workflow (when the metadata were not discarded, but committed into a version control system).

The local-only workflow is similar to the approach described in this article. However, we extend it:

- to support various kinds of metadata, not only information obtained using dynamic analysis
- and to support also line-level granularity by using source code comments.

Additionally, we will describe the approach in more detail, especially from the point of view of individual IDEs. Finally, we also provide a preliminary implementation of the approach, which was not available so far.

II. Source File Overwriting Approach

Now we will conceptually describe our approach in a high-level manner. The workflow begins with the clean source code of a program, without any metadata present.

A. Annotation-Based Metadata

Let us consider a situation when a programmer wants to analyze the program using a tool which outputs results for each class, method, member variable or another annotable element in the source code. He runs a tool, independent of a particular IDE. The tool overwrites the existing source code files, writing an annotation above each relevant element. The resulting source code has this form (Java syntax):

```
@Metadata{value}
source_code_element
```

The `@Metadata` annotation must be unique to the given type of analysis and not present in the clean source code.

For example, if a tool produces a numeric metric for each method, a hypothetical source code snippet looks like this:

```
@SomeMetric{1.7}
public void method() {
  ...
}
```

Suppose the developer inspected the source code along with the metrics and modified it with respect to some task. Now he no longer needs the metrics to be displayed – for example, he starts to work on another task, or he wants to commit his changes to a version control system (VCS). There are two possibilities:

- He manually executes the tool, instructing it to remove the annotations from the source code (e.g., by command-line arguments).
- The tool is running in the background, waiting for the VCS command to start. When it starts, the tool pauses the VCS process, removes the annotations automatically and then resumes the process.

Either way, the source code submitted to a VCS will be clean and co-workers will not be aware of the programmer using the tool.

B. Comment-Based Metadata

Sometimes, method-, class- or member-level granularity is not enough. In case we need to assign line-level metadata, we can use one-line source code comments instead of annotations. To ensure the comments can be unambiguously removed after they are no longer necessary, they must have a specific format, not present in the clean, unannotated code. In general, lines labeled with metadata have this form:

```
code_line //special_character(s) metadata
```

For instance, suppose we want to display code coverage results. We need to assign a piece of metadata to each executable line of code: whether a line was executed or not. Then a possible source code snippet can look like this:

```
public void method() {
  int i = 1; //+ executed
  if (i == 2) { //+ executed
    doSomething(); //+ not executed
  }
}
```

After the metadata are no longer needed by the programmer, they can be removed in the same way as in the case of Java annotations: either manually or as an automatic reaction to a VCS process execution.

III. Tool Description

A preliminary implementation of our approach is named SOverwrite. The source code of the tool is available online[1].

A. Possibilities

The current version of SOverwrite includes three sample analysis types to demonstrate our approach:

- static analysis: callers of methods,
- dynamic analysis: code coverage,
- version control system: authors of last class/method modifications.

[1] http://bit.do/SOverwrite

1) Method Callers: The "method callers" sub-tool statically analyses the source code and inserts the Java annotation named `@Caller` above each method which has exactly one caller, for example:

```
@Caller{clazz=Math.class, method="square"}
public void multiply(int a, int b) {
  ...
}
```

If a method has multiple callers, they are grouped using the `@Callers` annotation:

```
@Callers{
 @Caller{clazz=Mat.class, method="square"}
 @Caller{clazz=Mat.class, method="fact"}
}
public void multiply(int a, int b) {
  ...
}
```

2) Code Coverage: The "code coverage" sub-tool executes the program and counts how many times each line was executed (using an instrumentation agent). To each line which was executed at least once, it appends a specially-formatted comment representing the execution count of this particular line. A sample snippet looks like this:

```
public void method() {
  int i = 0; //+1
  while (i < 2) { //+3
    doSomething(); //+2
    i++; //+2
  }
}
```

In the example, the second line was executed once, the third three times, etc.

3) Last Author: The "last author" sub-tool reads the version control system information (in our case, Git). Above each class and method, it inserts the Java annotation `Author` containing a name of the last author who modified any line of the class/method. An example follows.

```
@Author("John Doe")
public void method() {
  ...
}
```

B. Tool Usage

SOverwrite contains a set of configuration files which can be used for customization.

First, the programmer configures the location of the project to work with (its root directory). If some kind of dynamic analysis is going to be performed, a run-time configuration, such as the location of the executable and command-line arguments, is also necessary.

The user can also select which types of analysis (sub-tools) will be active. The analysis can be run on the whole source code tree or just for the selected parts: It is possible to select classes and methods to be labeled with metadata, while the other ones will remain untouched.

Then, the developer executes SOverwrite to perform the selected analysis kinds and label the source with desired metadata.

Finally, the tool currently does not support the automated removal of metadata before a VCS commit. Therefore, the developer manually triggers the removal command in SOverwrite when he considers it is necessary.

C. Implementation Details

Each analysis kind is implemented as a subclass of the class `AnalyseProcessor`. Therefore, it is possible to add new sub-tools to SOverwrite. When practical, the analysis is performed in parallel on multiple files to speed up the process.

To obtain VCS information, we used the JGit[2] library. For static analysis and source code modification, the libraries Roaster[3] and Spoon [9] were used. Finally, dynamic analysis is possible thanks to the Java agent instrumentation API (application programming interface).

IV. Evaluation

A brief evaluation of SOverwrite regarding the unwanted differences in files and automatic file reloading in IDEs follows.

A. Differences in Files

If the metadata are written into the source code and subsequently removed from it, all files should remain the same as before performing the analysis (suppose the developer did not modify them meanwhile).

If we call the labeling process $label$ and the removal process $remove$, the following should hold regarding the source code ($code$):

$$remove(label(code)) = code$$

Generally, SOverwrite fulfills this property. However, there is one exception – if annotation-based metadata are used, the imports of annotation classes remain in the file. We consider this an implementation issue and not an inherent property of the approach.

B. Automatic Reloading

It is convenient to trigger an analysis while the project of interest is open in an IDE. Therefore, a file which is currently displayed in the IDE is modified by another process. This can possibly lead to confusion if the file is not synchronized. The user will not see the results of the analysis before closing and reopening the given file, which would limit the usability of the tool.

Fortunately, many IDEs and text editors offer a possibility to automatically reload a file if it is modified by another

TABLE I. FILE-RELOADING CAPABILITY OF SELECTED IDES AND TEXT EDITORS

IDE or text editor	Manual file reloading	Automatic file reloading	
		supported	default
Eclipse	supported	yes	off
IntelliJ IDEA	supported	yes	off
NetBeans	supported	yes	off
jEdit	supported	yes	on
SciTE	supported	yes	on
Vim	supported	yes	off

process. We summarized the possibilities and behavior of selected popular Java IDEs and text editors in Table I.

We can see that both manual (on-request) and automatic (in background) file reloading is available in all surveyed applications. Furthermore, in some text editors, automatic file reloading is turned on by default. When the default option is off, it can be easily adjusted in the application's settings. Therefore, we can conclude that file reloading is not an issue at all.

V. Discussion

We will now discuss the limitations of our approach and its applicability to programming languages other than Java.

A. Limitations

The most prominent limitation of our approach is its usability only for purely textual metadata. Insertion of graphics, such as plots, diagrams, etc., is inherently not possible in its current version. However, Schugerl et al. [10] present an Eclipse plugin SE-Editor which displays images directly in the IDE code editor – in places where specially-formatted comments are inserted in the source code (e.g., `/*** http://image.url */`). If these two approaches were combined and SE-Editor was implemented also for other IDEs and text editors, it would effectively mean our approach could support also graphical elements.

Another practical limitation is the non-interactivity of the approach. Standard IDE plugins offer rich interaction possibilities. In contrast to them, the inserted annotations and comments are generally static and non-navigable. However, there are some exceptions. For example, if a Java annotation contains a parameter of type "class" (e.g., `Math.class`), it is clickable in many IDEs. For instance, in IntelliJ IDEA, clicking on a class name while holding the Ctrl key opens the given class in the IDE.

B. Applicability to Other Languages

Although SOverwrite is implemented in Java, our approach is applicable to almost every programming language.

Annotation-based labeling requires the given language to support attribute-oriented programming. Examples of such languages are Java and C#.

Comment-based labeling requires the given language to have one-line comments which can be appended at the end of lines. The range of languages supporting one-line comments is very broad.

VI. Related Work

Source code annotations are traditionally designed to be processed by tools automatically. For example, an annotation above a member variable can denote its correspondence to an XML element [11]. On the other hand, concern annotations [12] and specially formatted tags [13] contain information useful mainly for humans. However, both of them are usually shared between developers and committed to a VCS. In contrast to them, our approach utilizes annotations only temporarily, in one developer's workspace.

Lee et al. [14] designed a generic infrastructure for framework-specific IDE extensions. They try to reduce authors' work when creating an IDE extension for a new object-oriented framework. However, providing an infrastructure for multiple different IDEs is out of the scope of their work.

Asenov et al. [15] present an IDE which enables programmers to query and modify programs using a combination of source code and other resources. Custom queries can be written using scripts, and the queries are composable using pipes. While they try to clear the line between plugin authors and application developers, their approach is currently limited to a single IDE which they created from scratch. In contrast to them, we try to reuse existing IDEs as much as possible.

Juhár [16] discusses how to integrate concerns with IDEs using concern-oriented projections. However, such approaches are usually not IDE-independent.

In [17], we presented a semi-automated approach AutoAnnot to assign concern annotations to methods. More specifically, we tagged methods in the source code with feature tags such as `@NoteAdding` or `@FileSaving`. This was achieved using software reconnaissance [18] which computes a difference of sets of executed methods between two runs. Compared to AutoAnnot, SOverwrite is much more general and supports also comment-based labeling.

As we already mentioned, Schugerl et al. [10] present a plugin to display images in the source code view. This could be used to extend the possibilities of SOverwrite.

VII. Conclusion and Future Work

In this article, we presented an approach of program comprehension and analysis tools independent of a specific IDE. While the source code is open in an IDE, it is modified by a tool – metadata like analysis results are written directly into the code in a form of annotations or comments. Therefore,

[2] https://www.eclipse.org/jgit/

[3] https://github.com/forge/roaster

the tool is not aware of the IDE and the programmer can use an environment or text editor of his or her choice. When the metadata are no longer necessary, they are removed to prevent VCS pollution.

A sample implementation, SOverwrite, was described in the article too. We demonstrated how the source code can be labeled with the results of static and dynamic analysis, along with VCS information.

The sub-tools (analysis types) currently implemented in SOverwrite act only as a demonstration. In the future, we would like to perceive it as a framework to which tool writers can add custom features.

The system should be also optimized and performance evaluation should be performed. Assessing the impact on developers' work would be a useful future work direction.

Acknowledgment

This work was supported by project KEGA 047TUKE-4/2016 Integrating software processes into the teaching of programming.

This work was also supported by the FEI TUKE Grant no. FEI-2015-23 Pattern based domain-specific language development.

References

[1] M. Sulír and J. Porubän, "Labeling source code with metadata: A survey and taxonomy," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, pp. 721–729.

[2] T. Karrer, J.-P. Krämer, J. Diehl, B. Hartmann, and J. Borchers, "Stacksplorer: Call graph navigation helps increasing code maintenance efficiency," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 217–224.

[3] D. Röthlisberger, M. Härry, W. Binder, P. Moret, D. Ansaloni, A. Villazón, and O. Nierstrasz, "Exploiting dynamic information in IDEs improves speed and correctness of software maintenance tasks," *Software Engineering, IEEE Transactions on*, vol. 38, no. 3, pp. 579–591, May 2012.

[4] R. Holmes and A. Begel, "Deep Intellisense: A tool for rehydrating evaporated information," in *Proceedings of the 2008 International Working Conference on Mining Software Repositories*, ser. MSR '08. New York, NY, USA: ACM, 2008, pp. 23–26.

[5] A. W. Bradley and G. C. Murphy, "Supporting software history exploration," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, ser. MSR '11. New York, NY, USA: ACM, 2011, pp. 193–202.

[6] D. Beyer and A. Fararooy, "DepDigger: A tool for detecting complex low-level dependencies," in *Program Comprehension (ICPC), 2010 IEEE 18th International Conference on*, Jun. 2010, pp. 40–41.

[7] F. Beck, O. Moseler, S. Diehl, and G. Rey, "In situ understanding of performance bottlenecks through visually augmented code," in *Program Comprehension (ICPC), 2013 IEEE 21st International Conference on*, May 2013, pp. 63–72.

[8] M. Sulír and J. Porubän, "Exposing runtime information through source code annotations," *Acta Electrotechnica et Informatica*, vol. 17, no. 1, pp. 3–9, Apr. 2017.

[9] R. Pawlak, M. Monperrus, N. Petitprez, C. Noguera, and L. Seinturier, "SPOON: A library for implementing analyses and transformations of Java source code," *Software: Practice and Experience*, vol. 46, no. 9, pp. 1155–1179, 2016.

[10] P. Schugerl, J. Rilling, and P. Charland, "Beyond generated software documentation – a Web 2.0 perspective," in *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on*, Sep. 2009, pp. 547–550.

[11] S. Chodarev, "Development of human-friendly notation for XML-based languages," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2016, pp. 1565–1571.

[12] M. Sulír, M. Nosáľ, and J. Porubän, "Recording concerns in source code using annotations," *Computer Languages, Systems & Structures*, vol. 46, pp. 44–65, Nov. 2016.

[13] M. Storey, J. Ryall, J. Singer, D. Myers, L.-T. Cheng, and M. Muller, "How software developers use tagging to support reminding and refinding," *Software Engineering, IEEE Transactions on*, vol. 35, no. 4, pp. 470–483, Jul. 2009.

[14] H. Lee, M. Antkiewicz, and K. Czarnecki, "Towards a Generic Infrastructure for Framework-Specific Integrated Development Environment Extensions," in *Domain-Specific Program Development 2008*, J. Lawall and L. Réveillère, Eds., Nashville, United States, 2008.

[15] D. Asenov, P. Müller, and L. Vogel, "The IDE as a scriptable information system," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2016. New York, NY, USA: ACM, 2016, pp. 444–449.

[16] J. Juhár, "Integrating concerns with development environments," in *Companion Proceedings of the 2016 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity*, ser. SPLASH Companion 2016. New York, NY, USA: ACM, 2016, pp. 6–8.

[17] M. Sulír and J. Porubän, "Semi-automatic concern annotation using differential code coverage," in *2015 IEEE 13th International Scientific Conference on Informatics*, Nov. 2015, pp. 258–262.

[18] N. Wilde and M. C. Scully, "Software reconnaissance: Mapping program features to code," *Journal of Software Maintenance: Research and Practice*, vol. 7, no. 1, pp. 49–62, 1995.

Could the Movie Be Cute? Understanding the User-Generated Word-of-Mouth by Implementing Text Mining Analysis on the Movie Market

Urszula Świerczyńska-Kaczor
The Film School in Lodz
ul. Targowa 61/63, 90-323
Lodz, Poland
uswierczynska@filmschool.lodz.pl

Małgorzata Kotlińska
Faculty of Management
University of Lodz
Lodz, Poland
mkotlinska@filmschool.lodz.pl

Abstract— **On the movie market, the user-generated word-of-mouth (WOM) impacts on the box office revenue and the span of the life-cycle of a film production. This article raises the question to what extent text mining can be useful tool in the process of understanding the consumers' experience and the evaluation of movie WOM. The results of the presented empirical study suggest that a text mining analysis of film reviews can offer both valuable insight about the viewer's vocabulary which indicates how consumers think about the movie, and insight about the general WOM valence. The managerial implication of the study is related mostly to the film promotion. In this article, the empirical study is presented within the broader context of other possible methods aimed at evaluating movie WOM.**

Keywords— text mining; online word-of-mouth; movie market

I. Introduction

Word-of-mouth (WOM) generated by viewers on film websites is one of the factors influencing the movie's success or failure on a particular market. WOM and its effect on box office performance, viewer satisfaction and the evaluation of the film quality have been investigated in the literature, although the number of studies referring to the movie market is still limited. In most studies, the researchers focus on the evaluation of WOM at the aggregated level: the general valence of WOM or the general sentiment of the post. In the presented study we take a different approach, focusing not only on the general meaning of WOM, but also on the specific terms used by viewers which constitute the emotional valence of WOM. Understanding the vocabulary (the exact terms) used by viewers can be helpful in the process of developing a strategy of promotion, especially on social media. Moreover, it may be also presumed that the identification of emotional terms used by viewers can serve as a reference point in a 'quick' evaluation of movie WOM. We based our study on the implementation of text mining tools in the analysis of film online user-generated reviews.

The article is structured as follows. In part I we discuss the characteristics of WOM, then – in part II - the relationship between the WOM and box office performance. In part III we point to the possible approaches of WOM measurement. The following section of the article (part IV) refers to the presentation of the conducted analysis for three selected film productions. As the vocabulary of film reviews can differ depending on the type of film, all analyzed films are similar considering the targeted audience and the genre. The article ends with the discussion and the conclusions.

The presented study is a part of broader research project, conducted in The Filmschool in Lodz: 'Badania statutowe, number WOSF/UPB/2017/09'.

II. Charactersitics of Word-of-Mouth on the Movie Market

Due to the Internet, the impact of moviegoer reviews on consumer behaviour seems to outweigh the impact of professional critic reviews (see, for example the study of [15]). Word-of-mouth generated by the moviegoer reviews can be characterized as a construct of different dimensions. The selected features of WOM are discussed below.

1. The personal 'word-of-mouth' versus 'word-of-click'

The entire volume of WOM consists of both personal and online talk about the product, although the ubiquitousness of the Internet has led to the situation that consumers express their opinions and seek for information mostly online. However, personal recommendation still has some effect on the viewer's behaviour, for example, the study of [18] referring to Polish moviegoer behaviour, indicated that friends' and family recommendation is still an important factor in the process of film selection.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

2. The 'pre-consumption' WOM versus 'post-consumption' WOM

Most studies in literature refer to the WOM as 'post – consumption' reviews of the product. However, on the Internet social space, the 'talk about the product' can embrace both the posts about how consumers experienced the product, and the posts about the consumer's intention to buy the product. The study of [17] indicated that the WOM which evaluated films (post-consumption WOM) and the WOM about intentions to watch the film (pre-consumption WOM), should be differentiated. Moreover, it seems that the pre-consumption WOM can be further categorized into two groups: as the WOM triggered by the film promotion and the WOM triggered by information spread by other Internet users. For example, the study of [1] pointed to the importance of WOM generated by trailers (therefore the film promotion), and the relations between the understanding of the movie from the trailer, the liking of it and the viewer's intention to generate WOM.

3. The building awareness WOM and the persuasive WOM

As [17] pointed, the WOM can influence brand awareness and persuade customers to purchase the product. Internet forums about the film reviews (for example: 'IMDb', 'Rotten Tomatoes', Polish 'Filmweb') have a 'push' feature (persuasive function), meaning that consumers who enter these forums are actively looking for film reviews, they are often aware of the particular film title, and in this case, the role of the WOM is to encourage consumers to watch a particular film production. On the other hand, other social communities, for example Twitter, and Facebook, have 'pull' features (building awareness), as these forums actively introduce the consumers to the news, including information about available films (see – [17]).

4. The global WOM versus the local WOM

Film viewers have become 'global' consumers – not only the national pool of reviews may impact domestic consumers' choice of film production. The differences between the date of the premiere of film productions on different geographical markets leads to the situation where consumers from a domestic market can access the user-generated reviews written before the national premiere (for example, Polish cinemagoers can read the reviews from other geographical markets about two weeks before the Polish opening weekend).

5. The volume of WOM to consider: the time and the sequels

In most cases professional film critics write their reviews around the time of the film release at the cinema, but the viewer-generated reviews appear at different phases of the whole film life-cycle: in the pre-released stage on the particular market (due to its availability on other markets or piracy), during the cinema window screening, and during the time when the movie is available in other distribution channels (DVDs, VoD). Sometimes, as in the case of 'old movies', the reviews can be written many years after the film premiere, for example, the film 'Casablanca', produced in 1942, still receives 'fresh' reviews on the well-known film website Rotten Tomatoes. The volume of WOM is expanding, not only due to time, but also to the connection with other products, such as sequels. For sequels, the spectrum of reviews may be extended as the potential viewer can also read the reviews of the previous films. Considering the amount of the sequels on film markets, the volume of the 'sequel' WOM may not be insignificant.

III. Word-of-Mouth as a factor influencing the number of film views

Although the movie market has been getting more attention from researchers and the pool of interesting surveys in the literature has been steadily growing, the understanding of factors influencing movie success can still be assessed as limited. The difficulty of forecasting on the movie market can be illustrated by looking at recent extreme box office performances - for example, in 2017, on the one hand there were unexpected failures such as the screening of 'Man Down' in London ([4]), and on the other hand we had the spectacular success for such titles as 'Bahubali 2' [2]. In literature, many factors are listed as possible determinants of box office revenue (the first week and the cumulative box office) ([5]; [3]; [11]);

1. Factors connected with product attributes, such as: production budgets, symbolism, genre, type of screenplay (as original screenplay/sequel/adaptation), famous actors – 'stars' in the film, the popularity of the director, the perceived quality of the movie by consumers, MPAA rating, running time (minutes);

2. Distribution-related factors, such as: market power of the distributor, release date – seasonality, competition from other productions screened at the same time, number of film prints, number of screens;

3. Information sources, such as: advertising expenditures, critics' rating, audience rating, film awards.

The study of [12] about Bollywood movies indicates that both the volume and valence of user-generated reviews have a positive relationship with box-office revenue. [19] analyzed WOM generated by moviegoers through the lens of the Prospect Theory, and this study highlighted reference dependence, loss aversion, and diminishing sensitivity in moviegoers' reactions to WOM. The researchers point out that the moviegoers' evaluation of the movie depends not only on the volume and valence of WOM, but also on consumers' expectations based on previously received WOM, and the intensive advertising could backfire as the promotion raised the consumers' expectations. The study of [17], which addressed the Twitter WOM about movie productions, led to the conclusions that the valence of WOM had an effect on movie sales, the effect of WOM depended on the number of followers of the WOM source (the effect was more significant for Twitter users having more followers), and the strongest effect on movie sales came from the pre-consumption posts.

So far, relatively few studies have investigated such aspects of movie WOM as the helpfulness of reviews or the WOM entropy. The researchers – [16] - investigated the helpfulness of the movie reviews in the context of emotional content of reviews. As to the WOM entropy – understood as the level of heterogeneity in the review text (the high level means the equally distributed sentiment in the reviews), [7] pointed to the moderating effect of WOM entropy on the WOM and movie box office sales.

An effect of WOM which is generally neglected in the studies on the relationship between WOM and the box office revenue, is the piracy. As [8] indicate, movie piracy has a cannibalization effect, and also a promotional effect understood as word-of-mouth generated due to the illegal watching of the film. The cited study showed that for a very few films, pre-release piracy could increase the revenue compared with the situation when the piracy occurs at release. Nevertheless, piracy always leads to the loss of revenue ([8]). The problem of illegal distribution of movies can be noticeable and significant on many markets, including the Polish market ([13], [14]). To sum up, the official statistics of film revenue - indicated in the evaluation of how WOM affects the revenue - only partly reflects the real market as part of the 'film views' statistics are hidden due to illegal film distribution.

IV. Different approaches for the evaluation of WOM on the film market

Some of the factors influencing box office revenue are 'purely quantifiable', for example: the number of screens or film running time (in minutes). The method of evaluation of other factors can vary in different studies as there is no single standard of measurement. For example, film awards can be evaluated quantitatively as the 'number' of film awards, but the evaluation can also include a more subjective aspect, such as the prestige of awards (e.g. prestigious international film festival awards vs local awards). A similar problem of evaluation refers to the 'popularity of directors' or 'fame of the actors', and also to the word-of-mouth (WOM) generated by viewers.

On the film market, word-of-mouth can be measured at different levels. In the first situation, the analysis is conducted at the level of the whole community of Internet users who evaluate a particular film production, and it is based on the whole pool of reviews written by different people ('the review' is the 'nod' in the analysis, not the 'reviewer'). In the second situation, the analysis can embrace the profiles of registered reviewers who allow Internet users access to their profile and 'rating history'. In this case, the analysis of spreading the WOM may be based at the individual level, and include new variables, for example 'liking the genre by the reviewer', 'the reviewer tendency to overrate or underrate film productions' (reviewer 'becomes' the nod in the analysis of WOM spreading). Some studies link these two approaches, relating the feature of review and the characteristics of reviewer (as for example, in the study of [17] which links the character of movie posts and the number of followers of the sender on Twitter).

The distinction between the level of analysis: review or reviewer, should be followed by the division between the method of research based on observation or on communication with respondents. In the category of studies based on communications, as an example, can be indicated by the study of [10], in which the researchers evaluated the likelihood of generating WOM and the positive WOM by asking respondents to assess the following statements: "I am likely to recommend this movie to others', "I am likely to talk to other people about this film", "I will recommend this movie to other people" and "I will say positive things about this movie". Another group of methods implemented in WOM analysis utilize the data gathered without the respondents' direct involvement (therefore - not based on a questionnaire, focus group interviews and other communication methods), and the further discussion in this article is focused on this group.

The word-of-mouth is most often evaluated on the basis of a 'general' audience rating: the 'stars – points' given by the reviewers. However, even a superficial analysis of user-generated reviews can lead to the conclusion that the reviewer's rating – 'points' - and the text of reviews are often inconsistent. First, there are cases when the text of a film review written by a particular viewer does not match the number of points/stars given: for example, the text is enthusiastic about the film production, but the reviewer gives 'only' average on the 'stars scale'. Secondly, it is easily noticeable that the same number of 'stars' are interpreted differently in different reviews: for example, some reviewers give 'three stars out of five' for films described in the review as 'good', the other reviewers give the same number of stars for the films which they assess in the text of the review as 'mediocre'.

An alternative to the 'points/stars' approach lies in in the manual assessment of reviews and categorizing them as 'positive', 'neutral/mixed' or 'negative'. However, this approach raises the problem with the volume of the user generated reviews (for example over 1000 reviews), and requirements that the resulting classification should not be biased by one judge's perspective. The limitation connected with manual classification can be overcome by the implementation of an automatic technique of classification. For example, in the study conducted by [19] or in the study of [17], the researchers classified the text of film reviews by constructing the classifiers from training sets. Although, the approach based on automatic classification allows researchers to analyze the large volume of WOM, it also has its disadvantages. For example, the method may lead to the possible inclusion of irrelevant posts, especially when the title of the film is also a commonly used expression, therefore this problem must be overcome in the research process (for example, see the study of [19], [17]).

The approaches based on 'points-stars' and manual or automatic classification lead to the evaluation of each analyzed review, and therefore, the pool of the reviews can be evaluated on the basis of quantified metrics, for example: the number of positive/negative reviews, the percentage of negative/positive reviews out of the pool of all reviews, the ratio of positive reviews to negative reviews and so on.

The next approach to the analysis of film reviews lies in text mining analysis. Text mining analysis delivers a 'quick overview' of the meaning and structure of textual data generated by the whole community. In this situation we are often more focused on the 'general picture' about the analyzed data rather than on the analysis of each post. It is worth emphasizing that the text mining analysis gives a 'quick inside look' about the analyzed subject, but the further and more detailed analysis often needs manual work ([6]). As an illustrative example of the implementation of text mining analysis on the movie market the study of [7] can be considered. To classify the reviews as positive, negative and neutral, [7] conducted a Latent Semantic Analysis (LSA). In the cited study, the text mining analysis was performed in the following steps: 1. Text preprocessing (calculating the number of words, number of characters, their variances, the summary statistics of the review text); 2. Parsing (text stemming, creation of term-by-frequncy matrix); 3. Term reduction (reduction of stop words, identification of synonyms, creation of Term Frequency-Inverse Document Frequency to place weight on more popular and less popular terms); 4. Singular value decomposition (reduction of dimensionality of the term-by-frequency matrix by grouping terms), 5. Text cluster analysis (leading to clustering each review text into categories: positive, negative or other, then 'other category' was manually distinguished to neutral or irrelevant).

V. Characteristics of movie WOM– text mining analysis for the selected films

Our study aims to capture the characteristics of movie WOM by the identification of the vocabulary – words, connotations - which are used by viewers to describe their film experience. For analysis we selected three films: 'Despicable Me 3', 'Baby Boss' and 'Zootopia'. The analyzed films share some similarities: they were released in 2016 and 2017, they target the same group of potential viewers – young children with their parents, and they are animations rated as PG – Parental Guidance Suggested. We took the samples of online film reviews from two major worldwide film websites: Rotten Tomatoes, and IMDb (Table I).

The audience film rankings of the analyzed films – based on 'points/stars' - indicate that the productions are evaluated differently by viewers. 'Zootopia' received the most favorable ratio of positive to negative reviews - with the ratio over 11 (the result is not surprising as the film was an Oscar winner in 2017 in the category 'animated feature film'). The 'Despicable Me 3' ratio is over 3, and 'Baby Boss' – the ratio is 1.6 (Table II).

Based on the Statistica and KH Coder software we conducted text mining analyses of film reviews (note - due to the volume of results, the presentation of the full spectrum of data is not possible here – and therefore, we present below only a selected scope of the conducted analysis and results).

- The text mining analysis led to the extraction 29 concepts for 'Despicable Me 3', 29 concepts for 'Zootopia' and 34 for 'Baby Boss' (Statistica software).
- The ranking of word-importance indicates that most words are connected with the story plot, and only a very few words can be related to possible 'positive' or 'negative' meanings: 'good', 'bad', 'like' (Statistica software).
- The emotional appeal of the film is much more visible in the ranking of 'word residuals'. Therefore, from the ranking of top 100 'word residual' of each analyzed film, we selected all terms with emotional valence: negative and positive, and in the next step - we identified the words which appeared in all three rankings. It led to the identification of 19 words (stems), such as: bad, bore, disappoint, fun, cute, laugh, like, best, enjoy, entertain, joke, funni, good, nice, perfect, fantast, hilari, love, awesome. Five more words were identified in the rankings of two film productions (Table III).
- For each film we identified the words most often connected with the words from the catalogue, for example, 'cute-movie', 'cute-funny', 'fun-family', 'fun-really'.
- We analyzed and visualized the structure of data using the graphical representation of data, for example Multi-Dimensional Scaling (KH Coder), and a screen plot of SVD word coefficients (Statistica Software).
- The words being predictors of viewers' ranking based on points have negative meaning for all three films, for example: 'worst', 'overr (overrated)', 'waste', 'bore' (Statistica Software).

Table I. Descriptive statistics – sample of audience reviews

Nr	Title	Sample	The procedure of obtaining sample Note – the date of writing posts by viewer can be different than the date of sample selection, only English reviews analyzed (other languages were excluded), apart from Zootopia, the reviews without direct 'stars' rating (e.g. reviews labelled as 'not interested') were excluded from analysis
1.	Despicable Me 3 (2017) Première: 14.06.2017	N=814	Rotten Tomatoes https://www.rottentomatoes.com and Internet Movie Database (IMDb) http://www.imdb.com/title/tt3469046/?ref_=tt_pg [29.08.2017], date of sample selection: 27-29.08.2017. As the websites use different 'stars' rating for films, we adjusted the rating of IMDb to match the scale of Rotten Tomatoes reviews with given points of film ranking.
2.	Baby Boss (2017) Première: 10.03.2017	N=365	https://www.rottentomatoes.com, date of sample selection: 16-23.04 2017
3.	Zootopia (2016) Première: 10.02.2016	N=702	https://www.rottentomatoes.com, date of sample selection: 16-23.04 2017; 4 reviews do not include 'stars' rating, but these 4 reviews were included in text mining analysis

VI. Conclusions

Marketers and film producers can gain insight about the consumer experience through the analysis of user-generated word-of-mouth (WOM). This article is focused on the specific aspect of WOM - understanding the vocabulary which is used by the viewers. The conducted text mining analysis of film reviews led to the identification of 19 terms with emotional valence which contribute to the meaning of reviews for different film productions. These identified terms - for example ‘fun’, ‘cute’, laugh’, ‘love’ - indicate the way consumers describe and therefore ‘think’ about the process of experiencing the movie (we can also answer the question in this article title – ‘Yes, the movie could be cute’). The identified terms can be helpful in the process of film promotion, especially on such channels as social media. There is also the question of whether, and to what extent, the identified words could be used or imbedded into the film poster tagline. Poster taglines, usually consisting of one or two short sentences, are an important element of promotion as they are one of elements framing the film (see the research of [9]).

Apart from promotion, the next aspect lies in the possible connection between identified terms and the evaluation of WOM as positive or negative. It may be presumed that the identified terms can be perceived as a kind of catalogue of ‘words’ and ‘connection’ which might be a ‘reference point’ for evaluation on the movie market. In the presented analysis of three movies, it can be noticed that the film with the highest evaluation on the basis of ‘stars/points’ – ‘Zootopia’ – is the production for which the positive terms are relatively high in the ranking of ‘word-residual’ (if the ranks of 16 positive words are summed up, ‘Zootopia’ has the lowest sum, followed by ‘Despicable Me 3’ and ‘Baby Boss’ – the result reflects the ‘stars’ evaluation). On the other side, for ‘Zootopia’ the negative terms have relatively low ranks and the collocation between the word ‘film’ - ‘bad’ is also low in ranking. Therefore, the information of which ‘reference terms’ are present in the ‘negative and positive ranking’ of a particular film production and which ‘reference terms’ are missing, might work as a ‘quick’ indicator of the valence of movie WOM.

The advantage of the presented approach lies in the

Table II. The analyzed customer-generated reviews: the audience film ranking calculated from the obtained samples

	Despicable Me 3 (2017)		**Baby Boss (2017)**		**Zootopia (2016)**	
	n	**%**	**n**	**%**	**n**	**%**
Stars 0.5-2.5 (negative)	167	20.5%	127	34.8%	55	7.9%
Stars 3 (neutral)	86	10.6%	33	9.0%	30	4.3%
Stars 3.5-5.0 (positive)	561	68.9%	205	56.2%	613	87.8%
Sum	814	100%	365	100%	698	100%
Ration: positive to negative reviews	3.4		1.6		11.1	

Table III. The words with the emotional valence and the rank of the word in the word residual ranking

	Despicable Me 3	**Baby Boss**	**Zootopia**		**Despicable Me 3**	**Baby Boss**	**Zootopia**		**Despicable Me 3**	**Baby Boss**	**Zootopia**
term	Positive words present within top 100 words in the ranking of word residual for each analyzed film			term	Words present within the ranking of word residuals, but not necessary within 100 top words			term	Words present for at least two films		
fun	35	2	1	funni	9	217	4	favorit	72	N	62
cute	3	5	13	good	168	75	22	amaz	5	N	27
laugh	97	23	74	nice	236	69	95				
like	55	74	8	perfect	25	210	41				
best	17	9	94	fantast	78	158	51				
enjoy	23	62	19	hilari	83	3	166				
entertain	19	33	18	love	105	36	3				
joke	50	49	72	awesom	6	93	138				
Word with negative meaning											
	Despicable Me 3	**Baby Boss**	**Zootopia**		**Despicable Me 3**	**Baby Boss**	**Zootopia**				
	Rank of the word in the word residual ranking			term	Words present for at least two films						
bad	16	4	204	silli	74	153	N				
bore	114	48	310	trribl	221	31	N				
disappoint	46	251	402	wast	184	95	N				

'quickness' of receiving the general outlook about a film production, without categorizing each review. Although the presented exploratory study has its limitations, for example the limited number of analyzed films, the possible bias of the review sampling. We can also intuitively assume that the vocabulary and the catalogue of words depend significantly on the film genre (e.g. within the genre 'horror', the movie could not be 'cute'). An interesting field of future studies refers to the question of how the extracted terms would be perceived in the promotion, for example on trailers or posters. Do the advertising slogans which directly refer to 'fun', 'cuteness' have better persuasive potential compared with other slogans? What can be the role of the indicated words in different forms of film promotions? To indicate further future research, it is also worth pointing to the studies which place the online movie reviews into the broader context of investigating emotions. Future studies should not only classify film reviews according to the positive or negative sentiment, but also investigate the connection between the WOM and the spectrum of emotions, for example: anger, enjoyment, distress or surprise. In this scope of research understanding the exact vocabulary used by reviewers in the process of reviewing films from different genres could be a crucial aspect of research.

The studies and literature about the opportunities and problems of the implementation of text mining within the scope of the film industry is still very limited. Unlike many other studies, our research highlights the valence of the viewers' terms and points to the importance of emotional content of reviews. This article also illustrates the potential problems with the evaluation of WOM - and therefore, the possible discrepancies between results obtained by usage of the different methods. In literature, there is no single standard metric implemented in the process of the measurement of WOM for a particular film production. The film viewer-generated reviews are usually evaluated on audience rating: the 'points' (stars) given by the viewers or on the manual or automatic evaluation of the reviews. As WOM can influence the film box office revenue, and its influence will grow due to the ubiquity of Internet, further research about the nature and metrics of WOM for film industry is interesting, and still an under-researched area.

References

[1] Archer-Brown C., Kampani J., Marder B., Bal S. A., Kietzmann J. (2017), Conditions in Prerelease Movie Trailers For Stimulating Positive Word of Mouth, Journal of Advertising Research Jun 2017, 57 (2) 159-172; DOI: 10.2501/JAR-2017-023

[2] BBC, 3 May 2017, Bahubali 2: South Indian epic film sees fans troll Bollywood, http://www.bbc.com/news/world-asia-india-39777539 [28.08.2017]

[3] Belvaux B., Marteaux S. (2007), Web user opinions as an information source. What impact on cinema attendances?, Recherche et Applications en Marketing, vol. 22, n° 3/2007

[4] Filmweb, http://www.filmweb.pl/news/Ile+os%C3%B3b+zobaczy%C5%82o+w+Wielkiej+Brytanii+nowy+film+z+LaBeoufem-122511)

[5] Gmerek N. (2015), The determinants of Polish movies' box office performance in Poland, Journal of Marketing and Consumer Behaviour in Emerging Markets 1(1)2015, DOI: 10.7172/2449-6634.jmcbem.2015.1.2

[6] Kayser V., Blind K. (2017), Extending the knowledge base of foresight: The contribution of text mining, Technological Forecasting & Social Change 116 (2017) 208–215, http://dx.doi.org/10.1016/j.techfore.2016.10.017

[7] Lee J. H., Jung S. H., Park J. (2017), The role of entropy of review text sentiments on online WOM and movie box office sales, Electronic Commerce Research and Applications, Volume 22, 2017, Pages 42-52, ISSN 1567-4223, https://doi.org/10.1016/j.elerap.2017.03.001.

[8] Ma L., Montgomery A. L, Smith M. D. (2016), The Dual Impact of Movie Piracy on Box-Office Revenue: Cannibalization and Promotion, February 24, 2016, Available at SSRN: https://ssrn.com/abstract=2736946 [25.08.2017]

[9] Mahlknecht J. (2015), Three words to tell a story: the movie poster tagline, Word & Image, 31:4, 414-424, DOI: 10.1080/02666286.2015.1053036

[10] Mishra P., Bakshi M., Singh R. (2016), Impact of consumption emotions on WOM in movie consumption: Empirical evidence from emerging markets, Australasian Marketing Journal (AMJ), Volume 24, Issue 1, 2016, Pages 59-67, ISSN 1441-3582, https://doi.org/10.1016/j.ausmj.2015.12.005.

[11] Moon S., Bergey P. K., & Iacobucci D. (2010), Dynamic Effects Among Movie Ratings, Movie Revenues, and Viewer Satisfaction, Journal of Marketing, Vol. 74 (January 2010), 108–121

[12] Niraj R., Singh J. (2015), Impact of user-generated and professional critics reviews on Bollywood movie success, Australasian Marketing Journal 23 (2015) 179–187, http://dx.doi.org/10.1016/j.ausmj.2015.02.001

[13] Report PwC Polska (2014), Analiza wpływu zjawiska piractwa treści wideo na gospodarkę w Polsce, Available at: https://www.pwc.pl/pl/publikacje/piractwo/analiza_wplywu_zjawiska_piractwa_tresci_wideo_na_gospodarke_w_polsce_raport_pwc.pdf [26.08.2017]

[14] Report Deloitte. (2017), Piractwo w Internecie – straty dla kultury i gospodarki. Analiza wpływu zjawiska piractwa internetowego na gospodarkę Polski na wybranych rynkach kultury, Warszawa 2017, Available at: https://www2.deloitte.com/pl/pl/pages/zarzadzania-procesami-i-strategiczne/articles/piractwo-w-internecie-straty-dla-kultury-i-gospodarki.html [26.08.2017]

[15] Tsao, W.-C. (2014). Which type of online review is more persuasive? The influence of consumer reviews and critic ratings on moviegoers. Electronic Commerce Research. 14. 559-583. 10.1007/s10660-014-9160-5.

[16] Ullah R., Zeb A., Kim W. (2015) The impact of emotions on the helpfulness of movie reviews, Journal of Applied Research and Technology, Volume 13, Issue 3, 2015, Pages 359-363, ISSN 1665-6423, https://doi.org/10.1016/j.jart.2015.02.001.

[17] Rui H., Liu Y., Whinston A. (2013), Whose and what chatter matters? The effect of tweets on movie sales, Decision Support Systems, Volume 55, Issue 4, 2013, Pages 863-870, ISSN 0167-9236, https://doi.org/10.1016/j.dss.2012.12.022.

[18] Wolny R., Jaciow M., Krężołek D., Gadomska M., Wawro A. (2015), Widz kinowy w Polsce. Raport z badań., Uniwersytet Ekonomiczny w Katowicach, Centrum Badań i Transferu Wiedzy Uniwersytet Ekonomiczny w Katowicach, Katowice, Sierpień 2015

[19] Yoon Y., Polpanumas C., Park Y. J. (2017), The Impact of Word of Mouth via Twitter On Moviegoers' Decisions and Film Revenues, Revisiting Prospect Theory: How WOM about Movies Drives Loss-Aversion and Reference-Dependence Behaviors, Journal of Advertising Research, 2017, vol. 57 no. 2 144-158, DOI: 10.2501/JAR-2017-022

Self-driving cars – the human side

Péter Szikora Ph.D., Nikolett Madarász
Óbuda University, Keleti Faculty of Business and Management, Budapest, Hungary
szikora.peter@kgk.uni-obuda.hu
madarász.nikolett@kgk.uni-obuda.hu

***Abstract*— The purpose of this study is to circle a quickly developing, therefore very exciting topic. Nowadays, self-propelled cars are not just opportunities of the future, but are an important part of our present. The article describes the steps and types of self-driving cars and their impact on people.**

I. Introduction

The usage and production of these cars has become a leading industry in almost every area of the world. The world's car stock exceeded 80 million after the Second World War, then more than 90 million in 1960. Five years later this number was 130 million, 291 million in 1980, 419 million in 1990, and 731 million in 2011. According to the forecasts, it will reach two billion by 2020 [1]. Over the years and centuries, this industry has gone through enormous development, as the first vehicles were only powered by steam engine, then petrol and diesel came to public mind and currently it seems that the electric propulsion will be the future. Of course, with this development, faster and more useful vehicles can be produced, but in our accelerated world with more and more cars, unfortunately the numbers of accidents have increased.

In most cases, these accidents are the fault of the driver, therefore it could be theoretically replaceable with the help of self-propelled cars. Human presence is the most important part in transport at present, although there are many areas where you can use a tool or feature that helps people achieve greater efficiency. Some examples for these features are the autopilot on aircraft, the cruise control in cars, and many other tools that help decision-making. In this study, we will provide a brief summary about the development of self-driving or at least driver assisting devices and show how people are feeling about this.

II. Evolution of Self-driving cars

Autonomous cars are those vehicles which are driven by digital technologies without any human intervention. They are capable of driving and navigating themselves on the roads by sensing the environmental impacts. Their appearance is designed to occupy less space on the road in order to avoid traffic jams and reduce the likelihood of accidents. Although the progression is gigantic, in 2017, allowed automated cars on public roads are not fully autonomous: each one needs a human driver who notices when it is necessary to take back the control over the vehicle [2].

The dream of self-propelled cars goes back to the Middle Ages, centuries before the invention of the car. An evidence for this statement comes from sketches of Leonardo De Vinci, in which he made a rough plan of them. Later, in literature and in several science fiction novels, the robots and the vehicles controlled by them, appeared.

The first driverless cars were prototyped in the 1920s, but they looked different than they are today. Although the "driver" was nominally lacking, these vehicles relied heavily on specific external inputs.

One of these solutions is when the car is controlled by another car behind it [3]. Its prototype was introduced in New York and Milwaukee known as „the American Wonder" or "Phantom Auto".

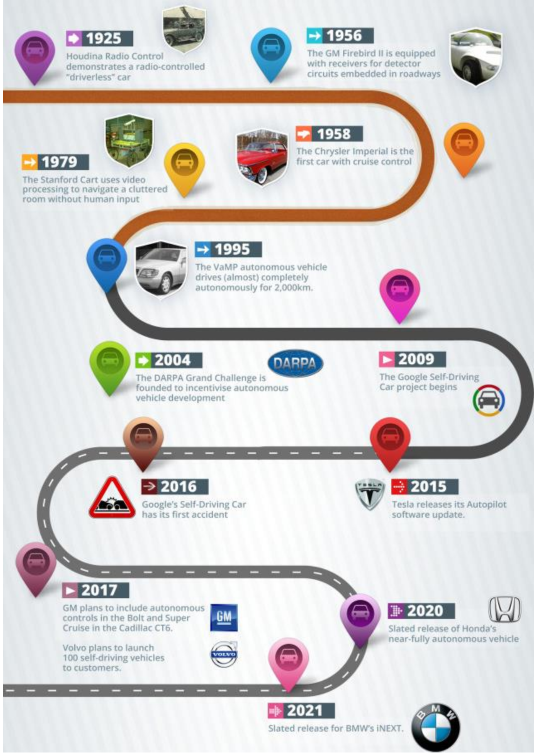

1925
Houdina Radio Control demonstrates a radio-controlled "driverless" car
1956
The GM Firebird II is equipped with receivers for detector circuits embedded in roadways
1958
The Chrysler Imperial is the first car with cruise control
1979
The Stanford Cart uses video processing to navigate a cluttered room without human input
1995
The VaMP autonomous vehicle drives (almost) completely autonomously for 2,000km.
2004
DARPA
The DARPA Grand Challenge is founded to incentivise autonomous vehicle development
2009
The Google Self-Driving Car project begins
2016
Google's Self-Driving Car has its first accident
TESLA
2015
Tesla releases its Autopilot software update.
2017
GM plans to include autonomous controls in the Bolt and Super Cruise in the Cadillac CT6.
GM
Volvo plans to launch 100 self-driving vehicles to customers.
VOLVO
2020
Slated release of Honda's near-fully autonomous vehicle
2021
Slated release for BMW's iNEXT.
BMW

Figure 1. – Evolution of self-driving cars [3].

Most of the big names – Mercedes Benz, Audi, BMW, Tesla, Hyundai etc. – have begun developing or forming partnerships around autonomous technology. They invested sizable resources into this, and by making this step they wanted to be leaders at the market of self-driving cars.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

Up to this point, numerous aids, software and sensors have been put into these cars, but we are still far from full autonomy.

They use lasers that are testing the environment with the help of LIDAR ((Light Detection and Ranging). This optical technology senses the shape and movement of objects around the car; combined with the digital GPS map of the area, they detect white and yellow lines on the road, as well as all standing and moving objects on their perimeter.

Autonomous vehicles can only drive themselves if the human driver can take over the control if needed.

These are those features that driverless cars already use [4]:

•Collision avoidance

•Drifting warning

•Blind-spot detectors

•Enhanced cruise control

•Self-parking

Below we briefly present some companies that play the most important role in the innovation of this segment, to show how this industry has developed.

Tesla [5]

Elon Musk, the Chief Executive Officer of Tesla, claims that every Tesla car will be completely autonomous within two years. The Tesla's "S" model is a semi-self-propelled car, where different cars are able to learn from each other while working together. The signals processed by the sensors are sent to other cars thus they can develop each other. This information teaches cars about changing lanes and detecting obstacles, and are continually improving from day to day. From October 2016, all Tesla vehicles have been being built by Autopilot Hardware 2, with a sensor and computing package that the company claims to allow complete self-driving without human interference.

Google [5]

The Google team has been working on driverless cars for years, and last year a working prototype was presented (by them). Furthermore, Google also supports other car manufacturers with self-driving car technologies such as Toyota Prius, Audi TT, and Lexus RX450h. Their own autonomous vehicle uses Bosch sensors and other equipment manufactured by LG and Continental companies. In 2014, Google planned a driverless car that would be available without pedals and wheels to make it available to the general public by 2020, but according to the current trends its fulfilment is still unlikely.

nuTonomy [5]

A Small group of graduates of the Massachusetts Institute of Technology (MIT) created the nuTonomy software and algorithm especially to self-propelled cars. In Singapore, nuTonomy has already put sensors to the Mitsubishi i-MiEV electric car prototype, thus nuTonomy algorithms can control the car on these complex urban roads by using GPS and LiDAR sensors. Besides that, in November 2016, they announced that self-propelled cars will be tested in Boston as well.

III. Types of Autonomous Vehicles

The National Highway Traffic Safety Administration (NHTSA) adopted the levels of the Society of Automotive Engineers for automated driving systems, which provides a broad spectrum of total human participation to total autonomy [6].

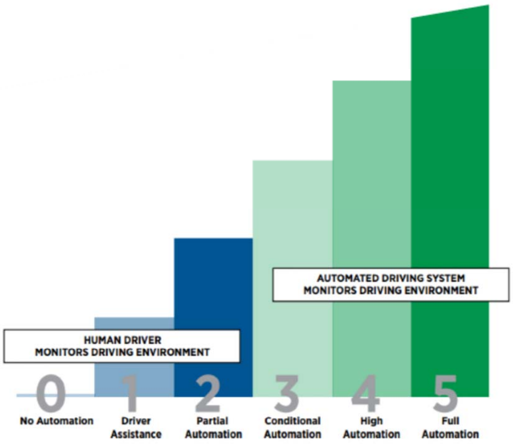


Figure 2. –Types of autonomous vehicles [21].

NHTSA expects automobile manufacturers to classify each vehicle in the coming years using SAE 0 to 5 levels.

These are the levels of SAE:

Level 0: No Automation [7]

In this case, there is 100% of human presence. Acceleration, braking and steering are constantly controlled by a human driver, even if they support warning sounds or safety intervention systems. This level also includes automated emergency braking.

Level 1: Driver Assistance [7]

The computer never controls steering and accelerating or braking simultaneously. In certain driving modes, the car can take control of the steering wheel or pedals. The best examples for the first level are adaptive cruise control and parking assistance.

Level 2: Partial Automation [7]

The driver can take his hands off the steering wheel. At this level, there are set-up options in which the car can control both pedals and the steering wheel at the same time, but only under certain circumstances. During this time the driver has to pay attention and if it is necessary, intervene. This is what Tesla Autopilot has known since 2014.

Level 3: Conditional Automation [7]

It approaches full autonomy, but this is dangerous in terms of liability, so therefore, paying attention to them is a very important element. Here the car has a certain mode that can take full responsibility for driving in certain circumstances, but the driver must take the control back when the system asks. At this level, the car can decide when to change lanes and how to respond to dynamic events on the road and it uses the human driver as a backup system.

Level 4: High Automation [7]

It is similar to the previous level, but it is much safer. The vehicle can drive itself under suitable circumstances, and it does not need human intervention. If the car meets something that it cannot handle, it will ask for human help, but it will not endanger passengers if there is no human response. These cars are close to the fully self-driving car.

Figure 3. - Types of autonomous vehicles [22].

Level 5: Full Automation [7]

At this level, as the car drives itself, human presence is not a necessity, only an opportunity. The front seats can turn backwards so passengers can talk more easily with each other, because the car does not need help in driving. All driving tasks are performed by the computer on any road under any circumstances, whether there's a human on board or not.

These levels are very useful as with these we can keep track of what happens when we move from human-driven cars to fully automated ones. This transition will have enormous consequences for our lives, our work and our future travels.

As autonomous driving options are widespread, the most advanced detection, vision, and control technologies allow cars to detect and monitor all objects around the car, relying on real-time object measurements. In addition, the information technology built into the vehicle is fully capable of delivering both external (field) and internal (machine) information to the car [4], [8].

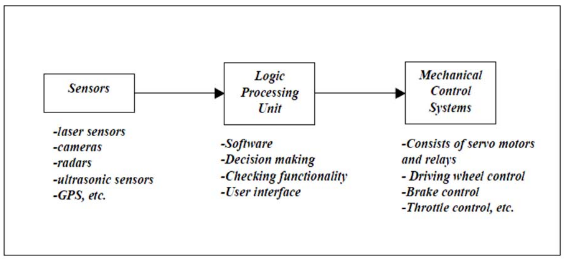


Figure 4. – System of self-driving cars [8].

IV. DECISIONS

Self-driving cars may be the future of transportation but we do not really know whether it is safer than non-autonomous driving or not. There are unexpected events during driving that force us to decide, often these are only tiny things such as passing through the yellow light or not but sometimes situations arise where we have to decide on the lives of others or our own.

Automakers are spending billions each year to develop self-propelled cars. But it turned out from different studies that people are more concerned than enthusiastic about the appearance of this new technology [9].

The University of Michigan made several researches, which systematically prove that drivers are concerned about fully autonomous cars, but they want some automated features.

Trusting in new technologies is expected to be a significant challenge for the public. Few people feel comfortable about using a new and unproven transportation technology, which can be seen after studying the aviation history. Although the Wright brothers flew with Kitty Hawk in 1903, the United States did not have a significant passenger airline until the1920s [10]. Similar concerns raised about the safety and reliability of the technology of self-propelled cars [11], [12], [13], [14]. Besides that, people are worried about giving full control to the system, and general beliefs also show that these cars are not (or not sufficiently) safe. Concerns about the safety and reliability of emerging technologies are natural and we want to study their effect on the enthusiasm for driving cars.

In 2011, the Accenture in the United States and the United Kingdom, made a research with 2 006 consumers. Nearly half of the respondents reported that they would be comfortable using driverless cars and the other half were more likely to use the technology if they could take control again if needed [15]. In 2012, J.D. Power and Associates conducted a survey of 17,400 vehicle owners; 37% said they would be interested in purchasing a fully autonomous car, but this number dropped 20% when they became aware of the costs [16].

Percentage of responses, by gender and age, to the question: “Vehicle manufacturers are considering using one of three levels of automation in future vehicles. Which level would you prefer to have in your personal vehicle? [17]”

Answers to this question are in the following table:

TABLE I. Results of the research [17].

Response	Gender		Age				Total
	Female	*Male*	*18-29*	*30-44*	*45-59*	*60+*	
No self-driving	48.4	43.1	41.3	35.2	50.6	56.2	45.8
Partially self-driving	39.8	37.5	39.9	42.6	37.8	34.2	38.7
Completely self-driving	11.8	19.4	18.8	22.2	11.6	9.6	15.5

The cars that drive themselves and the computer program that controls it will have to make difficult moral decisions in extreme situations. If a child runs out in front of the car, and the collision is unavoidable, will the car pull the steering wheel thus endangering the passengers, or rather hit the child on the road [18]?

Numerous researchers have been looking for the answer for the question how machines should make ethical decisions. For example, the Massachusetts Institute of Technology invented/created the Moral Machine that is made for gathering human perspectives in ethical questions in connection with self-driving cars.

These problems may arise:

- How should the car be programmed to act in the event of an unavoidable accident?
- Should it minimize the loss of life even if it means sacrificing the occupants, or should it protect the occupants at all costs?
- Should it choose between these extremes at random?

Answers to these ethical questions are important because they can have a great impact on the ability to accept autonomous cars in society. Who would buy a car that is programmed to sacrifice the owner [19]?

V. REASONS FOR FEAR

However, they have not only examined the causes of trust, but also their lack. The most important and most common causes are the following [18].

TABLE II. The reasons for fear [20]

When perceiving danger, it can put a steer lock on the truck	2.608
Electronics can easily be influenced	2.451
There is no firewall on the system	2.403
It block-breaks when on slippery road upslope	2.346
It breaks when on slippery road disregarding the load	2.320
When front sensors get wet only the truck is being braked and the trailer is forgotten	2.300
In case of electronic failure (most) brakes malfunction	2.205
When turned off it can opt to restart automatically	2.059
It is dangerous when with oversize load	2.058

These data are from a blink that was available online and also in a pen and paper format. 238 professional truck drivers have filled out the questionnaire. 188 male and 50 female [20].

In a September report, the National Road Safety Authority (NHTSA) said that "manufacturers and other entities working cooperatively with regulators and other stakeholders (e.g. Drivers, Passengers and Endangered Road Users) must deal with these situations to ensure that ethical considerations and decisions are made consciously and intentionally."

In a statement, the NHTSA said that the questions of public opinion in connection with the risks of self-driving cars on roads should be answered as accurately as possible and warned about the consequential corporate attitude of limiting liability. Public opinion will play a significant role in making self-driving cars widely accepted, so this is why it is important to examine ethics in these issues. These researches include ethical dilemmas to see how they respond.

The results of them are only the first attempt to develop automated robot-driven vehicles through moral algorithms raised by moral problems. These dilemmas are about unavoidable accidents where people have to decide on the outcome of the event, as we would expect from autonomous cars later on.

One way to approach this kind of problem is acting in a way that minimizes loss of life. According to this way of thinking, killing a man is better than killing other 10 people. But in this case, the consequences may be that fewer people will buy a car, driven by a robot, because cars would be prepared to sacrifice owners [19].

These ethical dilemmas have been raised to hundreds of workers on Amazon's Mechanical Turk to find out what they thought. Participants received outcomes in which one or more pedestrians could be rescued when the car crashed into an obstacle, killing its passenger or a pedestrian while saving other pedestrians.

Results are interesting if predictable. Usually people are satisfied with the idea that self-propelled vehicles should be programmed to reduce the number of victims. These results were true until the respondents looked at the question from the side of a pedestrian or an anonymous person, and their viewpoint suddenly changed when they were in the car.

Of course, this is not the only fear we have to deal with. Any computer that communicates with another computer or in some way accessible, are in danger of computer hackers. In recent years, there have been many news about hackers who broke into various databases, about viruses infected many computers, and there have been numerous terrorist acts where cars were driven into the crowd. Computer-driven vehicles, unfortunately, provide opportunity to accomplish these actions.

VI. CONCLUSIONS

The object of this document is to give insight to the international literature of autonomous cars. This is a quite new topic, and this is still closer to a piece of science fiction literature, but several companies try to solve this task, even if there are many problems with it.

We showed that the main problem is the fear of losing control. If a computer decides instead of us, we do not control the processes. Every computer and program may have a back door, and the question arising is what can be done if someone enters into the computer that can save our lives.

This is the beginning of my research, later we would like to deal with this issue, through my own results at first with questionnaire. Which includes problems related to fears; people's needs and expectations; and how they relate to self-driving cars etc.

References

[1] J. Voelcker, „1.2 Billion Vehicles On World's Roads Now, 2 Billion By 2035: Report," 2014. [Online]. Available: http://www.greencarreports.com/news/1093560_1-2-billion-vehicles-on-worlds-roads-now-2-billion-by-2035-report.

[2] D. Liden, „What Is a Driverless Car?," 2017. [Online]. Available: http://www.wisegeek.com/what-is-a-driverless-car.htm.

[3] U. Jenn, „The Road to Driverless Cars: 1925 - 2025.," 2016. [Online]. Available: http://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/12665/The-Road-to-Driverless-Cars-1925--2025.aspx.

[4] G. Gates, K. Granville, J. Markoff, K. Russell és A. And Singhvi, „The Race for Self-Driving Cars," 2017. [Online]. Available: https://www.nytimes.com/interactive/2016/12/14/technology/how-self-driving-cars-work.html.

[5] C. Mercer, „12 Companies Making Driverless Cars You Should Know About.," 2017. [Online]. Available: http://www.techworld.com/picture-gallery/data/-companies-working-on-driverless-cars-3641537/.

[6] H. Reese, „Updated: Autonomous driving levels 0 to 5: Understanding the differences.," 2016. [Online]. Available: http://www.techrepublic.com/article/autonomous-driving-levels-0-to-5-understanding-the-differences/.

[7] L. Blain, „Self-driving vehicles: What are the six levels of autonomy?," 2017. [Online]. Available: http://newatlas.com/sae-autonomous-levels-definition-self-driving/49947/.

[8] A. WIRED, „A Brief History of Autonomous Vehicle Technology.," 2017. [Online]. Available: https://www.wired.com/brandlab/2016/03/a-brief-history-of-autonomous-vehicle-technology/.

[9] Z. Enwemeka, „Consumers Don't Really Want Self-Driving Cars, MIT Study Finds.," 2017. [Online]. Available: http://www.wbur.org/bostonomix/2017/05/25/mit-study-self-driving-cars .

[10] J. E. Vance, Capturing the Horizon: The Historical Geography of Transportation., New York: Harper and Row, 1986.

[11] T. Economist., „Look, no hands.," 2013. [Online]. Available: http://www.economist.com/ .

[12] D. Neil, „Who's Behind the Wheel? Nobody.," 2012. [Online]. Available: http:// online.wsj.com/article/SB10000872396390443524904577651552635911824.html.

[13] K. Fitchard, „Ford is ready for the autonomous car. Are drivers?," 2012. [Online]. Available: http://gigaom.com/mobile/ford-is-ready-for-the-autonomous-car-are-drivers/.

[14] D. P. Howley, „The Race to Build Self-Driving Cars," 2012. [Online]. Available: http:// blog.laptopmag.com/high-tech-cars-go-mainstream-self-driving-in-car-radar-more.

[15] Accenture, „Consumers in US and UK Frustrated with Intelligent Devices That Frequently Crash or Freeze New Accenture Survey Finds," 2011. [Online]. Available: http:// newsroom.accenture.com/article_display.cfm?article_id=5146.

[16] L. M. Yvkoff, „Car buyers show interest in autonomous car tech.," 2012. [Online]. Available: http:// reviews.cnet.com/8301-13746_7-57422698-48/many-car-buyers-show-interest-in-autonomouscar-tech/.

[17] B. Schoettle és M. Sivak, „Motorists' Preferences for Different Levels of Vehicle Automation.," 2015. [Online]. Available: http://deepblue.lib.umich.edu/bitstream/handle/2027.42/114386/103217.pdf?sequence=1&isAllowed=y.

[18] G. Lubin, „Self-driving cars are already deciding who to kill," 2017. [Online]. Available: http://www.businessinsider.com/self-driving-cars-already-deciding-who-to-kill-2016-12.

[19] MIT, „Why Self-Driving Cars Must Be Programmed to Kill," 2017. [Online]. Available: https://www.technologyreview.com/s/542626/why-self-driving-cars-must-be-programmed-to-kill/.

[20] K. Lazányi, „DO YOU TRUST YOUR CAR?," IEEE 17th International Symposium on Computational Intelligence and Informatics, %1. kötet17, p. 11, 2016.

[21] Sae.org, „AUTOMATED DRIVING.," 2017. [Online]. Available: http://www.sae.org/misc/pdfs/automated_driving.pdf .

[22] Ghsa.org, „Autonomous Vehicles Meet Human Drivers: Traffic Safety Issues for States," 2017. [Online]. Available: http://www.ghsa.org/resources/spotlight-av17.

SAT-based Verification of NSPK Protocol Including Delays in the Network

Sabina Szymoniak
Inst. of Computer and Information Sciences
Czestochowa University of Technology
Czestochowa, Poland
sabina.szymoniak@icis.pcz.pl

Olga Siedlecka-Lamch
Inst. of Computer and Information Sciences
Czestochowa University of Technology
Czestochowa, Poland
olga.siedlecka@icis.pcz.pl

Mirosław Kurkowski
Institute of Computer Science
Cardinal St. Wyszynski University
Warsaw, Poland
m.kurkowski@uksw.edu.pl

***Abstract*—Time analysis of security protocols plays an important role in computer networks security. However, it has been mainly used in the form of timestamps analysis, without taking into account the parameters related to time.**

In this paper previous studies with use of a synchronized network of automata and SAT techniques have been extended with the temporal aspect and time parameters. A model was developed to show the strengths and weaknesses of the tested protocol depending on the known parameters of time. It has been shown that even potentially weak protocols can be used with appropriate time constraints. We can also find a way to make it safer by strengthening the critical points. As part of the work we have implemented a tool that helps us in the mentioned work and it allows to present some experimental results.

***Index Terms*—security protocols, modeling and verification, time analysis**

I. Introduction

Contemporary servers, terminals or other network communication devices use specially designed protocols to achieve important security objectives. Time analysis of those security protocols plays an important role. So far it has been mainly used in the form of timestamps without detailed analysis of time parameters.

Every conscious network user realizes that inside such protocols safeguards are included to ensure that data transmission will be safe - data will reach the destination and will not be decoded, or gained by dishonest user. On the other hand, networks administrators have increasingly difficult task, because the number of users and data is still growing. The data usually contain more and more sensitive information. The number of necessary encryption keys is growing and new protocols appear. Many parameters should be competently chosen: type of network protocol level of security, users' roles, etc. so that secure communication is available within a reasonable time.

So far, security protocols analysis was focused on one issue whether the Intruder can carry out the attack upon some honest user or the whole network. Using different verification methods based on formal, mathematical modeling i.e.: inductive [14], deductive [3] and discrete model checking [12], [13], [5], it was proven whether the considered protocol is correct and resistant to the attack. There are several high-profile projects linked with mathematical model checking of security protocols such as Avispa [1], ProVerif [2], SCYTHER [4] or native VerICS [8], [9], and PathFinder [17].

However, mentioned methods and tools usually ignore one, extremely important, parameter in their analysis - the time. Suppose that we have a simple three-step protocol and it was discovered that the attack upon this protocol can be executed in ten steps. It can be concluded that the protocol is not safe. However, the protocol can be secure using a time limit of communication for only three correct steps. Many protocols designers intuitively add timestamps to constructed protocols. However, how do they select timestamps without prior analysis?

In the work [10] a new formal, discrete, mathematical model of the protocols executions have been proposed thanks to which it is possible to prove correctness of time-dependent security protocol. This model was used to study the authentication processes. In Penczek and Jakubowska papers [6], [7] the network delays were taken into account. Their method was associated with time calculation of the proper communication session. Tested time constraints allow the indication of time influence on the protocol security. Mentioned studies of Penczek and Jakubowska involved only a single session and have not been continued.

In the presented work our previous studies with use of synchronized networks of automata and Boolean SAT techniques have been extended with time parameter and the temporal aspect. Developed model can show the strengths and weaknesses of the tested protocol depending on the known parameters of time. It has been shown that even potentially weak protocols can be used with appropriate time constraints - it can be secured by a suitable selection of time parameters. As part of the work we have implemented a tool that helps us in the mentioned research and allows to present some experimental results.

The rest of the paper is organized as follows. In the next section we show a protocol's example, basic concepts of formal language, and mathematical, computational structure used for formal modeling of protocol's executions. Also a simple example of formal describing of NSPK Protocol will be presented. Next, we give research assumptions and experimental results for timed versions of the original NSPK Protocol and Lowe's modification. The last section presents

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

our conclusions.

II. Formal language and computational structure

This section shows a basic concepts of forthe mal model and computational structure used for next investigations. For this purpose, we presentthe Needham Schroeder Public Key Protocol (NSPK Protocol) as an example.

The NSPK Protocol was proposed by Roger Needham and Michael Schroeder in 1978 [11]. The main goal of this protocol is mutual authentication of both users using specially generated pseudorandom numbers called *nonce* (it is shortcut from: *n*umber used *once*). A scheme of sending messages during the protocol execution (described in Common Language) is as follows:

$$\begin{array}{ll} 1. & \mathcal{A} \rightarrow \mathcal{B} : \langle \mathcal{N}_\mathcal{A}, \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{B}}, \\ 2. & \mathcal{B} \rightarrow \mathcal{A} : \langle \mathcal{N}_\mathcal{A}, \mathcal{N}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{A}}, \\ 3. & \mathcal{A} \rightarrow \mathcal{B} : \langle \mathcal{N}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{B}}. \end{array} \tag{1}$$

The NSPK Protocol consists of three steps. During it's execution two users $\mathcal{A}$ and $\mathcal{B}$ try to communicate, and authenticate to each other. In the first step the user $\mathcal{A}$ sends to the user $\mathcal{B}$ its nonce $\mathcal{N}_\mathcal{A}$ and it's identifier $\mathcal{I}_\mathcal{A}$ both encrypted by a public key of the user $\mathcal{B}$ $\mathcal{K}_\mathcal{B}$. The nonce $\mathcal{N}_\mathcal{A}$ is the pseudorandom big number specially generated by $\mathcal{A}$ only for this execution. Only the user $\mathcal{B}$ who has its own private key can decrypt the message and possess the nonce (a secret) $\mathcal{N}_\mathcal{A}$. Knowing it the user $\mathcal{B}$ sends to $\mathcal{A}$ nonce $\mathcal{N}_\mathcal{A}$ together with its own nonce $\mathcal{N}_\mathcal{B}$ both encrypted by a public key $\mathcal{K}_\mathcal{A}$. After this step the user $\mathcal{A}$ is sure that he is communicating with the user $\mathcal{B}$, because $\mathcal{B}$ have sent to $\mathcal{A}$ the secret $\mathcal{N}_\mathcal{A}$. In the last step the user $\mathcal{A}$ sends to $\mathcal{B}$ the nonce $\mathcal{N}_\mathcal{B}$ encrypted by a public key $\mathcal{K}_\mathcal{B}$. After this step the user $\mathcal{B}$ is sure that he is communicating with the user $\mathcal{A}$ because $\mathcal{A}$ have sent to $\mathcal{B}$ the secret $\mathcal{N}_\mathcal{B}$.

In 1995 Gavin Lowe with the use formal modeling and automated verification found an attack upon this protocol [12]. If we consider one more time, dishonest user of the network (denoted by $\mathcal{I}$) who is trying to deceive other users, we can find the following concurrent executions of NSPK:

$$\begin{array}{ll} \alpha_1. & \mathcal{A} \rightarrow \mathcal{I} : \langle \mathcal{N}_\mathcal{A}, \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{I}}, \\ & \quad \beta_1. \ \mathcal{I}(\mathcal{A}) \rightarrow \mathcal{B} : \langle \mathcal{N}_\mathcal{A}, \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{B}}, \\ & \quad \beta_2. \ \mathcal{B} \rightarrow \mathcal{I}(\mathcal{A}) : \langle \mathcal{N}_\mathcal{A}, \mathcal{N}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{A}}, \\ \alpha_2. & \mathcal{I} \rightarrow \mathcal{A} : \langle \mathcal{N}_\mathcal{A}, \mathcal{N}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{A}}, \\ \alpha_3. & \mathcal{A} \rightarrow \mathcal{I} : \langle \mathcal{N}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{I}}, \\ & \quad \beta_3. \ \mathcal{I}(\mathcal{A}) \rightarrow \mathcal{B} : \langle \mathcal{N}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{B}}. \end{array} \tag{2}$$

In this ececution the user $\mathcal{I}$ during communication with $\mathcal{B}$ impersonates the user $\mathcal{A}$. As we can see this scheme of NSPK adds two executions where the user $\mathcal{B}$ is cheated.

In paper [13] Lowe gave a corrected version of the protocol (called NSPK$_{Lowe}$) with the changed second step:

$$\mathcal{B} \rightarrow \mathcal{A} : \langle \mathcal{N}_\mathcal{A}, \mathcal{N}_\mathcal{B}, \mathcal{I}_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{A}}.$$

There is an additional identifier of the user $\mathcal{B}$ in ciphertext. This version is safe according to the Lowe's attack, and so far there is no other known attack upon this version.

In our consideration we investigate timed versions of mentioned before protocols. In this case nonces becomes into time tickets (timestamps). These tickets contain nonces and time moments of their generation. We will denote these timestamps by τ.

The main goal of this paper is to give computational structure in which we want to investigate time aspects of protocol's executions especially telated to delays in the network. For this purpose, we need a formal definition of a protocol that takes into account delays.

First, the definition of a protocol steps is required. Formally, every step of a timed protocol (taking into account network delays) is defined by two tuples α^1, α^2:

$$\alpha^1 = (S_{\rightarrow}, R_{\leftarrow}, L), \quad \alpha^2 = (\tau, D, X, G, tc). \tag{3}$$

In this notation the first tuple is similar to the previous method used for protocol's specification. It contains information about the sender, the receiver, and the messgae sent respectively.

The second tuple includes: X is the collection of letters necessary to construct the message, G is the collection of letters that must be generated by the sender in order to create the message. The second tuple α^2 consists of time aspects, τ denotes the moment the message was sent, D denotes the network delay, and tc is the set of temporal conditions that should be fulfilled to enable the protocol execution.

As an example, the formal definition of the timed version of the Needham Schroeder Public Key Protocol [11] is presented. The Protocol consists of three following steps:

$$\begin{array}{l} \alpha_1 {=} (\alpha_1^1, \alpha_1^2), \\ \alpha_1^1 {=} (\mathcal{A}; \mathcal{B}; \langle \tau_\mathcal{A} \cdot \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{B}}), \\ \alpha_1^2 {=} (\tau_1; D_1; \{\tau_\mathcal{A}, \mathcal{I}_\mathcal{A}, \mathcal{K}_\mathcal{B}\}; \{\tau_\mathcal{A}\}; \tau_1 + \mathcal{D}_1 - \tau_\mathcal{A} \leq \mathcal{L}_f). \end{array} \tag{4}$$

In the first step $\mathcal{A}$ and $\mathcal{B}$ denotes a sender and a receiver respectively. $\mathcal{I}_\mathcal{A}$ is the identifier of the user $\mathcal{A}$. $\tau_\mathcal{A}$ is the time ticket generated by the user $\mathcal{A}$. $\langle \tau_\mathcal{A} \cdot \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{B}}$ is the message sent in this step. The tuple α_1^2 includes the following temporal aspects - τ_1 denotes the moment the message was sent, D_1 denotes delay of the network, $\{\tau_\mathcal{A}, \mathcal{I}_\mathcal{A}, \mathcal{K}_\mathcal{B}\}$ is the collection of letters necessary to construct the message $\langle \tau_\mathcal{A} \cdot \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{B}}$. In this set we distinguish the set $\{\tau_\mathcal{A}\}$ containing the letter that must be generated by the sender $\mathcal{A}$ in order to create the message $\langle \tau_\mathcal{A} \cdot \mathcal{I}_\mathcal{A} \rangle_{\mathcal{K}_\mathcal{B}}$. $\tau_1 + \mathcal{D}_1 - \tau_\mathcal{A} \leq \mathcal{L}_f$ is the temporal condition that should be fulfilled to enable the protocol execution. In this case condition says that the value of the message sending time ($\tau_1 - \tau_\mathcal{A}$) plus value of the network's delay $\mathcal{D}_1$ should be less or equal to the lifetime $\mathcal{L}_f$.

$$\begin{array}{l} \alpha_2 {=} (\alpha_2^1, \alpha_2^2), \\ \alpha_2^1 {=} (\mathcal{B}; \mathcal{A}; \langle \tau_\mathcal{A} \cdot \tau_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{A}}), \\ \alpha_2^2 {=} (\tau_2; \mathcal{D}_2; \{\tau_\mathcal{A}, \tau_\mathcal{B}, \mathcal{K}_\mathcal{A}\}; \{\tau_\mathcal{B}\}; \\ \qquad \tau_2 + \mathcal{D}_2 - \tau_\mathcal{A} \leq \mathcal{L}_f \wedge \tau_2 + \mathcal{D}_2 - \tau_\mathcal{B} \leq \mathcal{L}_f). \end{array} \tag{5}$$

In the second step $\mathcal{B}$ and $\mathcal{A}$ similarly denotes the sender and the receiver. $\tau_\mathcal{B}$ is the time ticket generated by the user $\mathcal{B}$. $\langle \tau_\mathcal{A} \cdot \tau_\mathcal{B} \rangle_{\mathcal{K}_\mathcal{A}}$ is the message sent in this step. τ_2 denotes the moment the message was sent, D_2 denotes delay of the network, $\{\tau_{\mathcal{A}}, \tau_{\mathcal{B}}, \mathcal{K}_{\mathcal{A}}\}$ is the collection of letters necessary to construct the message $\langle \tau_{\mathcal{A}} \cdot \tau_{\mathcal{B}} \rangle_{\mathcal{K}_{\mathcal{A}}}$. The set $\{\tau_{\mathcal{B}}\}$ contains the letter that must be generated by the sender $\mathcal{B}$ in order to create the message. $\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{A}} \leq \mathcal{L}_f \wedge \tau_2 + \mathcal{D}_2 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$ is conjunction of two simple temporal conditions that should be fulfilled to enable execution of this step. This condition is similar to the previous one but takes into account the time τ_2 respectively.

$$\begin{aligned} &\alpha_3 {=} (\alpha_3^1, \alpha_3^2),\\ &\alpha_3^1 {=} (\mathcal{A}; \mathcal{B}; \langle \tau_{\mathcal{B}} \rangle_{\mathcal{K}_{\mathcal{B}}}),\\ &\alpha_3^2 {=} (\tau_3; \mathcal{D}_3; \{\tau_{\mathcal{A}}, \mathcal{K}_{\mathcal{B}}\}; \{\emptyset\};\\ &\qquad \tau_3 + \mathcal{D}_3 - \tau_{\mathcal{A}} \leq \mathcal{L}_f \wedge \tau_3 + \mathcal{D}_3 - \tau_{\mathcal{B}} \leq \mathcal{L}_f). \end{aligned} \tag{6}$$

The third step should be considered similarly.

Executions of security protocols can be modeled as specially designed discrete, mathematical, transition structures, for example using a synchronized network of timed automata. There are many types os synchronization of automata networks [15]. Our network works in the following way. The global state of the network is the tuple that consists of exactly one state from each automaton. The initial state of the network is global state that consists of all initial states of all automata. The global state labeled by α can be changed to the next one when all transitions in all automata labeled by α are enabled.

In our work we consider two types of timed automata which are included in the network. The first ones model protocols executions. The second type are automata that model users knowledge. Execution automata is modeling execution of an individual protocol step while preserving the imposed time conditions. Knowledge automata model the process of gaining knowledge by users. Knowledge and execution automata are synchronized with by labels, that allows to model the need to acquire spwcific knowledge by the user, so that he can execute next protocol steps..

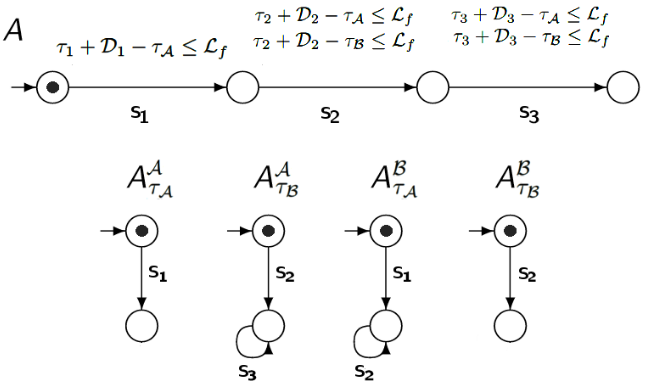

A
$\tau_1 + \mathcal{D}_1 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$
s1
s2
s3
$A^{\mathcal{A}}_{\tau_{\mathcal{A}}}$
$A^{\mathcal{A}}_{\tau_{\mathcal{B}}}$
$A^{\mathcal{B}}_{\tau_{\mathcal{A}}}$
$A^{\mathcal{B}}_{\tau_{\mathcal{B}}}$
s1
s2
s1
s2
s3
s2

Fig. 1. Network of synchronized timed automata for NSPK protocol - initial state.

The network for one execution of the NSPK protocol is shown on Fig. 1 (we have marked the global initial state of the network). The automaton A models the execution of all protocol step and time conditions. The next automata model changes the user's knowledge. For example an automaton $A^{\mathcal{A}}_{\tau_{\mathcal{A}}}$ models gaining knowledge about it's timestamp $\tau_{\mathcal{A}}$ by the user $\mathcal{A}$.

This global state can be changed to the next one what is shown on Fig. 2. Observe that the first transition in the automaton A is synchronized with the first transitions from automata $A^{\mathcal{A}}_{\tau_{\mathcal{A}}}$, and $A^{\mathcal{B}}_{\tau_{\mathcal{A}}}$. This is because the execution of the first step of the NSPK Protocol needs to gaining knowledge of the user $\mathcal{A}$ about the timestamp $\tau_{\mathcal{A}}$ (generating) and the user $\mathcal{B}$ about this ticket (possesing).

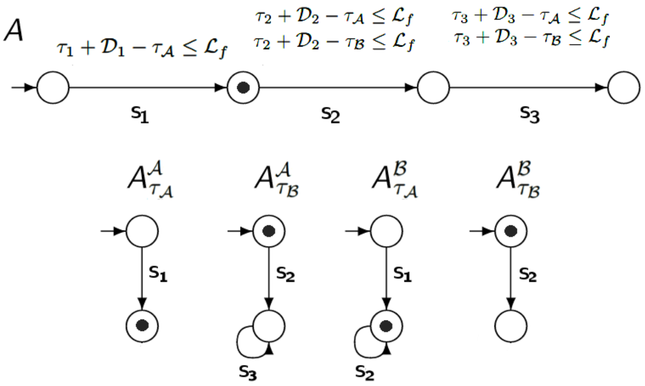

A
$\tau_1 + \mathcal{D}_1 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$
s1
s2
s3
$A^{\mathcal{A}}_{\tau_{\mathcal{A}}}$
$A^{\mathcal{A}}_{\tau_{\mathcal{B}}}$
$A^{\mathcal{B}}_{\tau_{\mathcal{A}}}$
$A^{\mathcal{B}}_{\tau_{\mathcal{B}}}$
s1
s2
s1
s2
s3
s2

Fig. 2. Network of synchronized timed automata for NSPK protocol - first step.

The second step of the protocol can be now executed (Fig. 3). See that the second transition in the automaton A is synchronized with the loop in the automaton $A^{\mathcal{B}}_{\tau_{\mathcal{A}}}$ (the user $\mathcal{B}$ cannot execute this step without knowledge about the ticket $\tau_{\mathcal{A}}$) and first transitions in the automata $A^{\mathcal{B}}_{\tau_{\mathcal{B}}}$ ($\mathcal{B}$ generates ticket $\tau_{\mathcal{B}}$ in this step), and $A^{\mathcal{A}}_{\tau_{\mathcal{B}}}$ ($\mathcal{A}$ posseses $\tau_{\mathcal{B}}$).

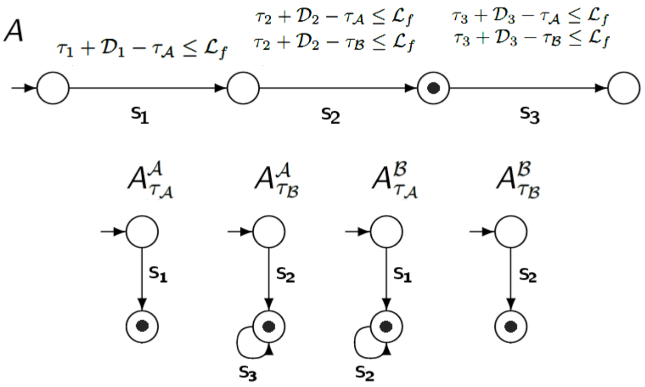

A
$\tau_1 + \mathcal{D}_1 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_{\mathcal{A}} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_{\mathcal{B}} \leq \mathcal{L}_f$
s1
s2
s3
$A^{\mathcal{A}}_{\tau_{\mathcal{A}}}$
$A^{\mathcal{A}}_{\tau_{\mathcal{B}}}$
$A^{\mathcal{B}}_{\tau_{\mathcal{A}}}$
$A^{\mathcal{B}}_{\tau_{\mathcal{B}}}$
s1
s2
s1
s2
s3
s2

Fig. 3. Network of synchronized timed automata for NSPK protocol - second step.

Finally the third step of the protocol can be performed. In Fig. 4 we can see that the third transition in the automaton A is synchronized with the loop in the automaton $A^{\mathcal{A}}_{\tau_{\mathcal{B}}}$ because the user $\mathcal{A}$ needs this knowledge for third step execute on.

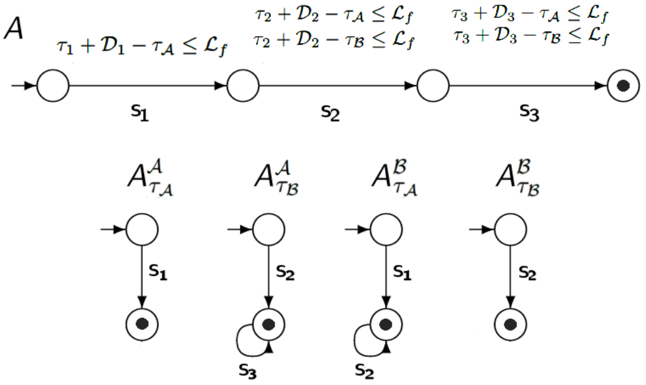

$\mathcal{A}$
$\tau_1 + \mathcal{D}_1 - \tau_\mathcal{A} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_\mathcal{A} \leq \mathcal{L}_f$
$\tau_2 + \mathcal{D}_2 - \tau_\mathcal{B} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_\mathcal{A} \leq \mathcal{L}_f$
$\tau_3 + \mathcal{D}_3 - \tau_\mathcal{B} \leq \mathcal{L}_f$
s_1
s_2
s_3
$\mathcal{A}^\mathcal{A}_{\tau_\mathcal{A}}$
$\mathcal{A}^\mathcal{A}_{\tau_\mathcal{B}}$
$\mathcal{A}^\mathcal{B}_{\tau_\mathcal{A}}$
$\mathcal{A}^\mathcal{B}_{\tau_\mathcal{B}}$
s_1
s_2
s_1
s_2
s_3
s_2

Fig. 4. Network of synchronized timed automata for NSPK protocol - third step.

The method of automatic generation of all automata that models executions in the investigated protocol and considered space of the users is given in [9].

III. SAT TESTING

According to the presented assumptions, a special tool has been implemented [18]. This tool is used for automatic modeling and security protocols executions generation including delays in the network.

The tool starts with the question to the type of tests to be performed. It is possible to choose from the following options:

1) research on the availability of selected states using SAT-solver,
2) analysis of the duration of executions,
3) simulations.

In the first case, the SAT-solver input files will be properly prepared. In the case of selecting the second and third types of tests, the result will be text files containing information about the time parameters and effects of the generated protocol executions. Some instructions made during the operation of the tool are common to all types of tests.

The upper limit of the timeout value (for SAT-solver tests) or network delay (for time analysis and simulation) should be given first. After that, set of all protocols participants are prepared and their initial knowledge is collected. In addition to the simulation tests, the probability distribution (homogeneous, normal, Poisson, Cauchy, exponential) must be selected. Then, network delay values will then be generated according to this distribution. In the next step is necessary to select the timed security protocol that will be loaded from the file (written in the ProToc specification language).

Depending on the selected type of tests, additional parameters of the tool operation are set at this stage. For analysis of executions times, selecting the type of execution that will be analyzed is necessary. Types of executions are as follows: all, attacking, the executions in which the intruder is, or the executions in which the intruder impersonates honest user. For simulation it is necessary to calculate timeouts in the individual steps of the protocol. After preparation of the parameters of the tool operation, the collections of the execution of the protocol and the chains are generated.

In the last step network of synchronized timed automata (automata of knowledge and execution) is generated. Next, this network id encoded into a Boolean formula that will be fed into the SAT-solver input. In turn, for the analysis of executions time and simulation, in the last step the protocol executions are generated. It is also possible to perform multiple analyzes for different protocols without restarting the tool. All operations performed include the writing of the relevant data to the files.

In the case of SAT-solver testing the network of automata is encoded into a Boolean, propositional, logical formula. Tested security property is also encoded into the such formula. The problem of fulfilling the property by the encoded system (set of protocol executions) is thereby extruded in a way that is mutually unambiguous to the problem of the satisfiability of the conjunction of the received logic of formulas (SAT problem). From the computational point of view these formulas are large. They have at least a few hundred variables, and their conjunction normal form contains thousands (hundred of thousands) of clauses.

It is well-known that problem of the formulas satisfiability is an NP-complete problem. In spite of this, the formulas obtained in the above manner can usually be tested using specialized tools called SAT-solvers. It is worth pointing out that with the unambiguous encoding method it is known that if there is a validation satisfying the given formula, then the set of executions meet the property under test. If we examine the suitability of a protocol set for an attack, then it is equivalent to the existence of an attack on the protocol.

The main parameters of this verification method are the length of the path, the number of used variables and clauses, the amount of memory used by the processor, and the calculation time. The tests were performed using a computer unit with the Linux Ubuntu operating system, processor Intel Core i7 and 16 GB RAM.

The study began with the selection of the appropriate SAT-solver. Five SAT solves were tested: Glucose, Lingeling, Treengeling, Clasp and MiniSAT. The summary of SAT-solver performance is as follows (Tab. I).

TABLE I
EXPERIMENTAL RESULTS FOR SELECTED SAT-SOLVER AND PROTOCOLS

Protocols	**MiniSAT**	**Lingeling**	**Clasp**	**Glucose**	**Treen-geling**
Memory [Mb]					
$NSPK$	2,95	2,99	3,01	3,2	4
$NSPK_{Lowe}$	3,95	4,4	4,02	4,23	5
Time [ms]					
$NSPK$	48	56	60	200	260
$NSPK_{Lowe}$	360	400	370	560	430

As can be inferred from the Tab. I, the best choice was the use of MINISAT.

The first tested protocol was the timed version of mentioned before Needham Schroeder protocol (NSPK). The synchronized network of timed automata for this protocol consisted of eighteen automata of executions and seventeen automata of knowledge (for honest users and Intruder). These studies dealt with occurrences run, in a given network, that terminated in one of the states modeling attacking execution. For a path of length 6, Lowe's attack [12] was discovered.

The first phase of the NSPK study was designed to demonstrate its vulnerability to attack with a delay in the network equal to 0,15 time unit ([tu]) and a lifetime of 2 [tu].

TABLE II
EXPERIMENTAL RESULTS FOR NSPK PROTOCOL

Length of path	Variable	Clauses	Memory [MB]	Time [ms]	Result
3	7322	18829	2,19	8	UNSAT
4	13873	34298	2,48	20	UNSAT
5	15051	37188	2,55	28	UNSAT
6	21849	53246	2,95	48	SAT

The Tab. II shows test results for the NSPK protocol. The table takes into account the relevant values of variables, clauses, memory and time, obtained on each path length, as well as an information whether a Boolean formula was satisfied or not at a given level. UNSAT annotation means that the formula has not been satisfied, and the SAT annotation that the formula is satisfied, so there is an attack upon this protocol.

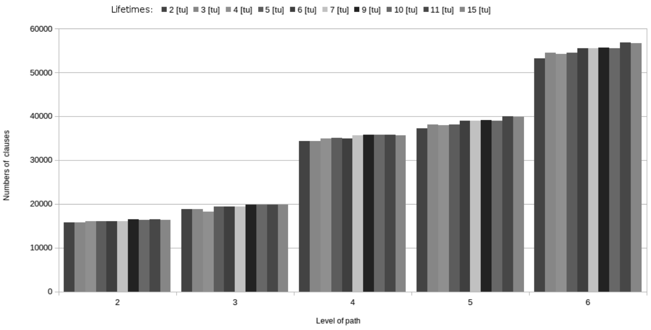


Fig. 5. Numbers of clauses for lengths of path

Fig. 5 shows the changes in the number of the clauses used for every path level for ten values of lifetime. The number of the clauses increased in each test series in proportion to each path length. In each test series other network delay values, as well as the lifetime value, were randomly generated. The delay in the network was randomly scaled from 0 to 1 [tu], while the lifetime value varied from 2 to 15 [tu].

The next phase of the NSPK protocol study involved an indication of the range of network delay values for which the protocol remains safe. This means that for given lifetime value a SAT-solver returns a value of UNSAT and there is no attack upon the protocol.

We have generated 1000 test series for different network delays and lifetime values. Based on the results obtained, it was determined that the protocol is vulnerable to attack if the delays satisfies the following conditions:

$$\mathcal{D}_1 \leq \mathcal{L}_f, \quad \mathcal{D}_2 \leq \frac{\mathcal{L}_f}{4}, \quad \mathcal{D}_3 \leq \frac{\mathcal{L}_f}{5}. \tag{7}$$

Thanks to such defined dependencies, it is possible to adjust time conditions in particular steps of the protocol. Correctly selected dependencies will make the honest user have enough time to complete the step and the Intruder (in the same time) could not gain additional knowledge to carry out the attack.

Considering the timed version of the NSPK [10], examining this protocol for network delay has shown that a more complex formula is needed in this case. For studies involving network delays, for track length 6, the number of variables was 24% higher and the number of clauses was 22% higher than in Kurkowski's case studies [10]. We do not compare the memory used by the processor and the duration of the test, as the research conducted by Kurkowski and the authors of this research took place on other systems and computer units.

Then, we conducted the tests taking into account the value of the time parameters in accordance with the designated dependencies. For the NSPK protocol lifetime was 10 [tu] and delays value were 10,1 [tu], 2,6 [tu] and 2,1 [tu] in the following steps.

TABLE III
EXPERIMENTAL RESULTS FOR NSPK PROTOCOL (WITH TIME CONDITION)

Length of path	Variable	Clauses	Memory [MB]	Time [ms]	Result
3	7533	19366	2,18	12	UNSAT
4	14177	35054	2,48	32	UNSAT
5	15431	38130	2,55	28	UNSAT
6	22334	54435	2,95	80	UNSAT

The Tab. III shows test results for the NSPK protocol with time conditions. Notice that the values of clauses and variables have increased. Moreover, for the chosen time parameters the tested formula was not satisfied. This means that there is no attack upon the NSPK protocol for this value of clocks.

Analogous tests were performed for the $NSPK_{Lowe}$ protocol [13]. This protocol is perceived to be secure, so we are only considering here the possibility of a "man in the middle" attack. For this protocol, the synchronized network of timed automata consisted of eighteen executions automata and nineteen knowledge automata. First, the vulnerability on attacks of this protocol was shown with a delay equal to 0.07 [tu] and a lifetime equal to 5 [tu].

As is shown in Tab. IV on a path of length 6 the man-in-the-middle attack was detected.

TABLE IV
EXPERIMENTAL RESULTS FOR NSPKL PROTOCOL

Length of path	Variable	Clauses	Memory [MB]	Time [ms]	Result
3	14540	35884	2,48	28	UNSAT
4	22864	55660	2,93	64	UNSAT
5	32210	77910	3,45	152	UNSAT
6	41445	99865	3,95	360	SAT

Test series for various network delay and various lifetime enabled us to find the range of network delay values for which the protocol remains safe. Thanks to the selected values and results obtained, it was determined that the Intruder would not be able to perform a man-in-the-middle attack if in delays were satisfied the following dependencies:

$$\mathcal{D}_1 \leq \mathcal{L}_f, \quad \mathcal{D}_2 \leq \frac{\mathcal{L}_f}{4}, \quad \mathcal{D}_3 \leq \frac{\mathcal{L}_f}{6}. \tag{8}$$

TABLE V
EXPERIMENTAL RESULTS FOR NSPK$_{Lowe}$ PROTOCOL (WITH TIME)

Length of path	Variable	Clauses	Memory [MB]	Time [ms]	Result
3	14751	36421	2,18	28	UNSAT
4	23168	56416	2,48	64	UNSAT
5	32590	78852	2,55	152	UNSAT
6	41930	101054	2,95	360	UNSAT

The Tab. V shows test results for the $NSPK_{Lowe}$ protocol with lifetime equal to 10 [tu], delays equal to 10,1 [tu], 2,6 [tu] and 1,7 [tu] in the following steps. The values of clauses and variables have increased again and tested formula was not satisfied. There is no attack (even man-in-the-middle) upon the $NSPK_{Lowe}$ protocol for this value of clocks.

IV. CONCLUSIONS

In this paper we have presented the results of formal, mathematical modeling and verification of NSPK protocol including delays in the computer network. To do this we have used discrete, automata based structures and propositional Boolean encoding.

Our computational results we have carried out using our tool and SAT-solver MiniSAT. These tests were designed to check the validity of the tested formula, and thus the attacks on the two well-known protocols, taking into account delays in the network. Appearance of delays in the formal model also enabled the extension of the study range. Therefore, security protocols could be considered in a broader sense, namely the study of the network's transmission time effects on Intruder's capabilities. In addition, the research has made it possible to verify that generated executions can actually occur. The conducted research has shown that for each protocol there is a validation satisfying the test formula. So, for every tested protocol, an attack was found.

The results show that properly selected methods and limitations effectively prevent the Intruder from attacking the tested protocols. Properly selected lifetime values make the protocol safer. Our further research will broaden the protocol analysis by combining the probability of capturing secret information [16] and time constraints.

REFERENCES

[1] Armando, A., et. al., *The AVISPA tool for the automated validation of internet security protocols and applications*, In: Proc. of 17th Int. Conf. on Computer Aided Verification (CAV'05), vol. 3576 of LNCS, pp. 281–285, Springer Verlag, 2005.

[2] Blanchet B., *Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif.* Foundations and Trends in Privacy and Security, vol. 1(1-2), pp.1–135, 2016.

[3] Burrows M., Abadi M., Needham R., *A Logic of Authentication.* In: Proceedings of the Royal Society of London A, vol. 426, pp. 233–271, 1989.

[4] Cremers, C., *The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols*, In: Proc. of the 20th International Conference on Computer Aided Verification, Princeton, USA, pp. 414–418, 2008.

[5] Dolev, D. and Yao, A., *On the security of public key protocols.* In: IEEE Transactions on Information Theory, 29(2), pp. 198–207, 1983.

[6] Jakubowska G., Penczek W., *Modeling and Checking Timed Authentication Security Protocols*, Proc. of the Int. Workshop on Concurrency, Specification and Programming (CS&P'06), Informatik-Berichte 206(2), pp. 280-291, Humboldt University Press, 2006.

[7] Jakubowska, G., and Penczek, W., *Is your security protocol on time?* In Proc. of FSEN'07, vol. 4767 of LNCS, pp. 65–80. Springer-Verlag, 2007.

[8] Kacprzak M., et. al., *Verics 2007 - a Model Checker for Knowledge and Real-Time*, Fundamenta Informatiace, vol. 85, pp. 313–328, IOS Press, 2008.

[9] Kurkowski, M., Penczek, W., *Applying Timed Automata to Model Checking of Security Protocols*, in ed. J. Wang, *Handbook of Finite State Based Models and Applications*, pp. 223–254, CRC Press, Boca Raton, USA, 2012.

[10] Kurkowski, M., *Formalne metody weryfikacji wlasności protokołów zabezpieczajacych w sieciach komputerowych*, Exit, Warsaw. 2013.

[11] Needham, R. M., Schroeder, M.D., *Using encryption for authentication in large networks of computers*. Commun. ACM, 21(12), 993–999, 1978.

[12] Lowe, G., *An Attack on The Needham-Schroeder Public-Key Authentication Protocol*, Information Processing Letters, vol. 56, No 3, pp. 131 - 133, 1995.

[13] Lowe, G. *Breaking and Fixing the Needham-Schroeder Public-key Protocol Using FDR*, In:TACAS, LNCS, Springer Verlag, pp. 147–166, 1996.

[14] Paulson L., *Inductive Analysis of the Internet Protocol TLS*, ACM Transactions on Information and System Security (TISSEC), vol 2 (3), pp. 332–351, 1999.

[15] Penczek, W., Półrola, A., *Advances in Verification of Time Petri Nets and Timed Automata: Temporal Logic Approach*, Studies in Computational Intelligence, vol. 20, Springer Verlag, 2006.

[16] Siedlecka-Lamch, O., Kurkowski, M., Piatkowski, J., *Probabilistic Model Checking of Security Protocols without Perfect Cryptography Assumption*, in Proc.: 23rd International Conference, Computer Networks 2016, Brunow, Poland, Communications in Computer and Information Science, vol. 608, pp. 107–117, Springer Verlag, 2016.

[17] Siedlecka-Lamch, O., Kurkowski, M., Piech, H., *A New Effective Approach for Modelling and Verification of Security Protocols*, in Proc. of 21th international Workshop on Concurrency, Specification and Programming (CS&P 2012), pp. 191-202, Humboldt University Press, Berlin, Germany, 2012.

[18] Szymoniak, S., Siedlecka-Lamch, O., Kurkowski, M., *Timed Analysis of Security Protocols*, in Proc.: 37th International Conference ISAT 2016, Karpacz, Poland, Advances in Intelligent Systems and Computing, vol. 522, pp. 53–63, Springer Verlag, 2017.

Using Eyetracking to Analyse How Flowcharts Are Understood

Petr Šaloun
FEECS, VSB–Technical University of Ostrava, Czech Republic
Email: petr.saloun@vsb.cz

Martin Malčík
VSB–Technical University of Ostrava, Czech Republic
Email: martin.malcik@vsb.cz

David Andrešič
FEECS, VSB–Technical University of Ostrava, Czech Republic
Email: david.andresic.st@vsb.cz

David Nespěšný
FEdu, University of Ostrava, Czech Republic
Email: david.nespesny@ahol.cz

***Abstract*—Natural language is frequently not the ideal tool to describe a procedure. Its interpretation is difficult and the language itself frequently allows for imprecise expressions; therefore if a procedure complies with basic conditions, we use an algorithm to describe it. An algorithm is a precise procedure which solves a specific task and leads to a given solution. Flowcharts are frequently used to graphically represent algorithms. A flowchart is in fact a graphical notation which plainly and unequivocally describes the flow of operations and activities, their sequence and basic relations between them. An algorithm in the form which is understood and executable by a machine is called a programme.**

In this article, we measure the understanding of flowcharts through the method of tracking eye movement together with galvanic skin conductance (GSC). The aim of this work is to discover the differences in reading a simple flowchart by people educated in technical fields and humanities and find out if this difference affects the emotional response during exercise-solving. This difference will then be decisive for learning the ability to transform a problem/procedure into an algorithm and programme it into a working application.

On the basis of the research we carried out, we can say that understanding flowcharts is different for people with different learning styles, but the emotional experiences of both groups are identical and only differ in the intensity of experienced emotions.

***Index Terms*—eye tracking, skin electrodermal activity, flowchart, learning styles**

I. Introduction

Education content includes a lot of individual elements and a range of various complex relations between them. To find their bearings in this whole, people need to first familiarise themselves with its individual components and then to confront them with the architecture of the whole. In one moment, they have to focus their attention on a certain part of the education content in order to understand it, and at the same time to divert their attention from parts which are not important at the given moment [1].

The quality of a memory trace is influenced by processes of encoding, storage and retrieval of information [2]. Encoding means instilling information into memory, and it can work visually, acoustically or semantically. The more senses involved, the more effective it is. An important thing for the storage of information is its meaningfulness, i.e. its classification within a whole, a structure. Information which has a certain meaning for students is more easily recollected – it can be connected with their personal needs or with intense emotional experience, absorbed intentionally, systematically and in relation to practice. On the other hand, mechanically instilled information disappears from memory more quickly, which is manifested by the fact that a student has a problem to recollect and decode [1], [20].

An interesting relation is a relation memory emotion. Emotions influence memory because they have an impact on the early selection of incoming information, i.e. processes a student is unaware of. They also influence the next phase, which is conscious processing of information. In addition, negative emotions slow down the redirection of attention to a new topic. Research also shows that positive, as well as negative emotions of moderate intensity can enhance the process of retaining information; however, excessively strong emotions (nervousness, threat) block short-term memory. Generally speaking, emotions either facilitate or complicate the connection to other stored information [3].

We selected a flowchart to analyse the understanding of non-linear text. Flowchart diagram is widely used, and it is easily understood by non-programmers. The UML uses an activity diagram to show a workflow – a starting point, actions, decisions, splits and joins to show concurrent activities, and ending points. It's essentially a flowchart for a process or workflow, usually written using domain specific terms. Activity diagram is the fifth most used UML diagram [4], [5].

II. Algorithmization

Algorithmization is an abstract process, which in the context of our research is described in more detail in connection with data structures in a book by Niclaus Wirth [6]. It is a way through which a student moves from a problem assignment to a procedure that leads to the problems solution. The challenge lies in the precise formulation of the problem using a natural language. Depending on the level of knowledge of the student and their education, text problem definition may be supplemented with flowcharts, which represent the

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

graphical formalism of problem-solving procedure annotation; flowcharts are defined by ISO 5807:1985 [7].

III. Flowcharts

The oldest documented proposal of use of flowcharts in the field of engineering is considered to be the publication by Mr. and Mrs. Gilbreth Process charts from 1921 [8]. As industry engineers, both dealt with production management, production ergonomics and study of movement of workers during production in order to increase the production effectiveness by reducing workers movement without impairing their health.

In the course of the twentieth century, there were a number of proposals of the graphic form of flowcharts with different uses (1940 – Bell Telephone Laboratories, 1950 – Functional Flow Block Diagram (FFBD) developed by TRW, later used by NASA). In 1962, a book by W. Gosling [9] was published, called The Design of engineering, in which he brought the idea that a flowchart must include an initial state and a final state as well as the other states through which the system cycles. The 1970s saw a boom in the creation of graphic representations and creation of formalisms in the field of graphic modelling in software engineering. A flowchart can be considered as the basic graphical representation of an algorithm.

The main purpose of diagrams is to clearly represent the processing procedure so that reading, understanding, processing and memorising the information presented by the diagram are as fast as possible and errorless. Therefore, diagrams have to be cognitively efficient. A cognitively efficient diagram is the one which is quickly, simply and precisely processed by the human mind [10]. The task of the diagram is therefore to graphically represent the proposal of a data processing procedure, which is then executed using an automated process. Functionality is the main purpose characteristic of the diagram and its faultlessness is expected. During the phase of compiling and testing of the diagram, the cognitive quality of its graphic notation has a significant impact. Prolonged error-discovering and fine-tuning can be caused by incorrectly designed or incomplete graphic notation. The final diagram is then used not only for the actual processing but also as documentation, where its printed form is used [11].

IV. Learning styles and flowcharts understanding

The manner in which students navigate the text and process new information (curriculum) to use to solve tasks is based on the nature of their thinking, intelligence, metacognitive abilities and personality characteristics.

Experts characterise individual learning methods differently. Mare [12] states they involve learning tactics, learning strategies, learning styles, approaches to learning, styles of thinking, etc.

The stated terms outline not only the course of learning, but also explain why specific students learn in a specific way, reaching specific results. The author defines learning style as a sum of methods which a person uses to learn during a specific life stage in most pedagogical situations. He further claims that they are partially dependent on the study content, form on an innate foundation and develop by the interaction of internal and external factors. Learning tactic is selected by a student deliberately it comprises of practically organized methods, which together create a learning strategy. Learning strategy is characterised as a sequence of coherently ordered activities with the intention of reaching a study goal. Strategy guides students to decide which abilities they use and in which order. Learning strategies are being fixed, later automated and closely connected with the learning style. The relation between learning styles and learning strategies is following: learning style has a form of a learning metastrategy and stands above various learning strategies [10]. Students success of solving a problem is therefore dependent on both the learning strategy (plan) and the learning style, while strategies are easier to influence.

The learning method is closely connected with the nature of thinking. Experts provide e.g. abstract thinking, specific thinking, etc. and point out its close connection with other psychic cognitive and non-cognitive functions.

According to Nakonen [1], thinking can be deep or superficial, flexible or rigid, agile or slow, independent or dependent, convergent or divergent, original or conventional and more, which evidences the connection of thinking to personality traits.

V. The use of eyetracking for the analysis of flowchart understanding

During reading, eyes fixate when they gather information and move between points in saccades. Eye tracking is a non-invasive behavioural measure that provides information about the moment-by-moment processes in which readers engage during reading. Eye tracking enables researchers to objectively track and record what respondents are doing when completing a comprehension task, how long they are doing it, and in what ocular sequence [13].

In general, eye movements during watching a stimulus can be divided into two categories: fixations (fixation of gaze on a particular location) and saccades (movement of gaze to another location). Saccades take 20–50 msec to complete depending upon the length of the movement and supposing almost no visual information is extracted during eye movements [13]. Between saccades, eyes remain stationary for brief, about 200 msec periods of time called fixations. Since visual information is only extracted from a page during fixations, reading is similar to a slide show in which short segments of the page are displayed for approximately a quarter of a second [14].

The resulting series of fixations and saccades form a curve called scanpath. The majority of information from an eye is available during the fixation but not during saccades.

The finding of various search abilities is a rapidly evolving issue, in the context of our research see for example [15]. Measuring the cognitive impact of eyetracker in our context is discussed in [16], and in [21] respectively.

VI. Electrodermal response

Since the 1880s, when psychological factors related to electrodermal phenomena were first observed, electrodermal recording has become one of the most frequently measured and used biosignal in psychology. The major reason for its popularity is the ease of obtaining a distinct electrodermal response (EDR), the intensity of which seems apparently related to stimulus intensity and/or its psychological significance. Electrodermal recording is possible with inexpensive equipment, not only in the laboratory but also under less controlled field conditions [17].

VII. Material and methods, Research goal

A. Research goal

The aim of the work is to find the differences in students perceptions of a simple flowchart and the differences in experiencing emotions that can be measured during the problem solving. The determined difference is then decisive for learning and the ability to transform the problem/procedure to an algorithm, and to program a functioning application.

B. Respondents

Respondents included non-paid students of lifelong learning of the study program for teaching aged 20 to 35 years. They were university graduates majoring in technology and economy. Experiment was conducted with 18 people, but only 15 were evaluated due to technical and physiological reasons.

C. Tasks

Respondents were given three tasks whose solving required understanding of a simple flowchart (see 1) which concerned entering a PIN code on a mobile phone. Despite entering a PIN code on a mobile is currently a frequently repeated activity, depicting it in the form of a flowchart is not that common; therefore there is a little probability that any respondent previously saw this stimulus. The respondents included a single programmer, who can be expected to be routinely able to read such a flowchart.

The first question respondents had to answer asked what happens after entering PIN 0000. The second question asked what happens after the user enters PIN 1234 three times. The third question required the respondents to actively seek the correct PIN from the flowchart. It required both passive understanding of the flowchart, and also to actively discover the correct PIN value.

D. Research methods

During testing, respondents were given three tasks in the form of flowcharts, which represented different cognitive levels of work. After the launch of the experiment, students were shown a flowchart with a written question. The experiment was not finished until the student solved all tasks correctly. The progress of the experiment was recorded with eye tracking and a Galvanic Skin Conductance (GSC) record was made.

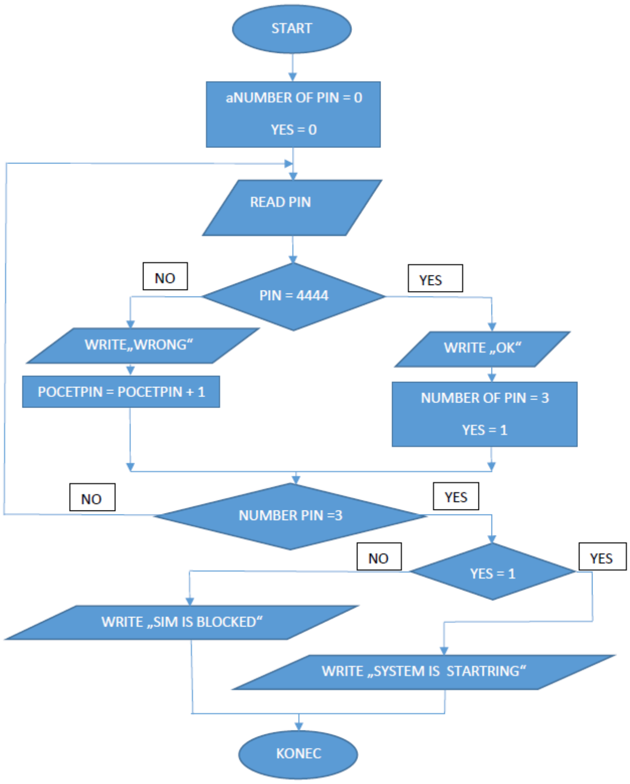


Fig. 1. Flowchart of a task. Source: EdLab authors.

E. Eye Tracking Technology

Eye tracking technology was used for the analysis of how students process flowcharts. Eye tracker Gaze Point GP3 was used to measure the number of eye movements and the length of eye fixations when respondents worked with flowcharts and Ogama Software Studio (version 5.0) was used for the analysis.

F. Skin conductance response and Education Laboratory Board (EdLab)

Skin conductance response is one of the most popular ways of measuring electrodermal activity. The EdLab measuring device is an interface for connecting up to 2 digital and 6 analogue sensors to a computer using a USB interface. EdLab uses a 12-bit A/D converter with 2×30 kHz sampling rate. EdLab software is used for recording and evaluating the measurement, which enables to export measured data in a CSV format. As a sensor, we used the sensor of skin conductance, which uses two electrodes placed on the fingertips (see 2) which work as terminals of a resistor (skin resistance). This resistance is the one line of the voltage divider. Voltage recorded in this way is filtered and increased and then evaluated by EdLaB measuring system through an equalising algorithm.

G. Results and discussion

We observed and analysed qualities (heat maps and gaze maps) and quantities (see below) in the eye tracking records.

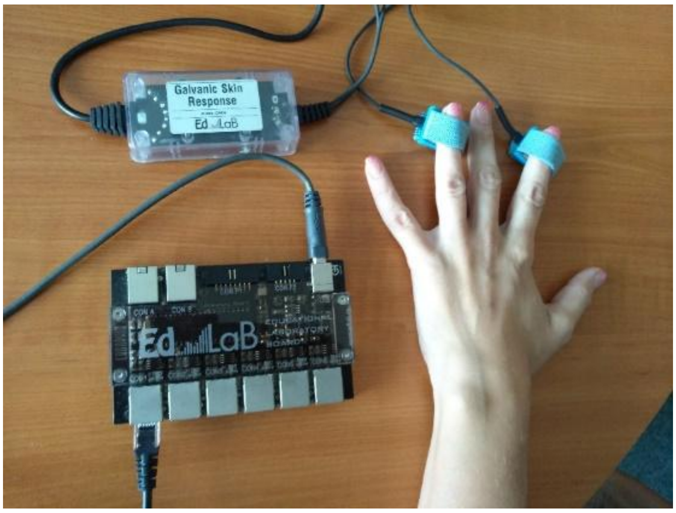


Fig. 2. Placement of sensors on respondents hand. Source: EdLab authors.

Concurrently, we analysed the galvanic skin conductance records of the respondents.

Tab. I shows several noteworthy facts in connection with the eye tracking records. Respondent R2 required the most time to solve the task, had the highest number of fixations, the longest Fixation Connections Length, but not the longest saccades. The quickest to solve the task was Respondent R3, who reached the shortest time with the smallest number of fixations, although their observation speed was the slowest. This conflict can be explained by the manner of the respondents work. They thought over some fixations (parts of the flowchart) and remained at those fixations for a longer time. Frequently, they would turn their eyes away from the monitor (stimulus) therefore the eye movement was not being tracked and the length of saccades increased.

Respondents R1 and R4 required a similar time to solve the task, but used different strategies. Respondent R1 focused their attention to the complete flowchart, although only the left block was needed to correctly solve the task. This respondent made many fixations and had a very short average length of saccades, which points out to a strategy focused on quick observation and search for connected information. From the learning style perspective, this is the serialist approach [19]. Compared to Respondent R1, Respondent R4 had very long saccades and few fixations, pointing out a contemplative and slower approach to information seeking for finding the correct answers. As far as learning style is concerned, this is the holistic approach [19].

H. Evaluating eye tracker measurements

The heatmaps of eye tracking records of individual respondents show several significant differences strategies of perception and understanding of the flowchart. All the Figures are related to the first task. Respondent on Figure 3 spread their attention relatively evenly across the complete flowchart, and rarely came back to the question in the top right corner.

Respondent R3 in Figure 4 dedicated a highly fragmented attention to the flowchart; the respondent focused solely on a few parts of it, ignoring the others. This respondent frequently turned their gaze back to the question in the top right corner, apparently having difficulty understanding or remembering the question.

Respondent in Figure 5 had previous experience with flowcharts, and only skimmed through individual branches and was quickly able to correctly answer the question.

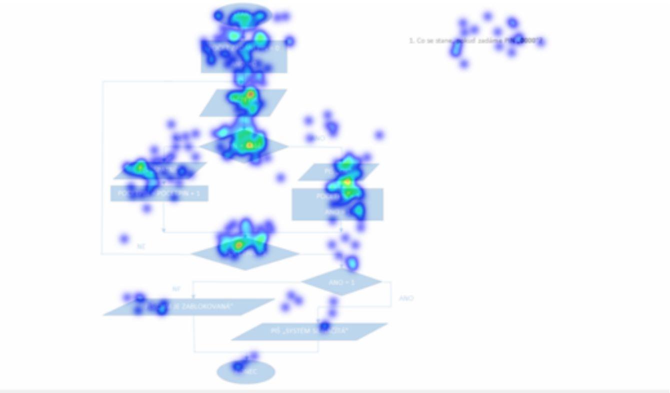


Fig. 3. Eye tracking heatmap of reading the flowchart by Respondent R1. Red and yellow regions indicate a higher frequency of users gazes.

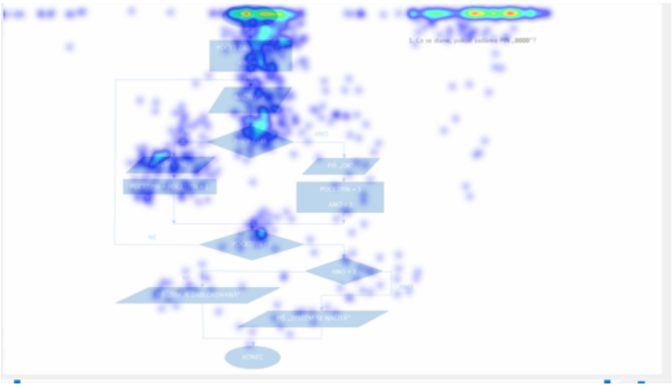


Fig. 4. Eye tracking heatmap of reading the flowchart by Respondent R2.

I. Evaluating GSC measurement data

GSC graphs exhibit several natural relations. Both graphs relate to Task 1. One group of respondents has shown an elevated initial emotional reaction at the beginning of problem solving and a gradual stilling of emotions (see 6). The other group has had a gradual rising of emotional exertion. Both groups have gone through several peaks, which approximately corresponded with finding the solution in the decision box in the flowchart.

Different graph monotonies (increasing, decreasing, or constant) evidence different approaches to the solving of the task and differently distributed attention during the problem solving. These differences are likely caused by different learning or personality styles of the respondents.

Name	Task Duration [msec]	Average Saccade Length [px]	Fixations [count]	Fixations/sec [count/sec]	Fixation Connections Length [px]	Fixation Duration Mean [msec]
R1	59932	146	338	5.6	49168	63.8
R2	143802	170	855	5.9	145188	71.0
R3	33055	248	96	2.9	23549	65.5
R4	57467	218	270	4.7	58610	74.5

TABLE I
EYE TRACKER PARAMETERS OF SOME RESPONDENTS SOLVING TASK 1.

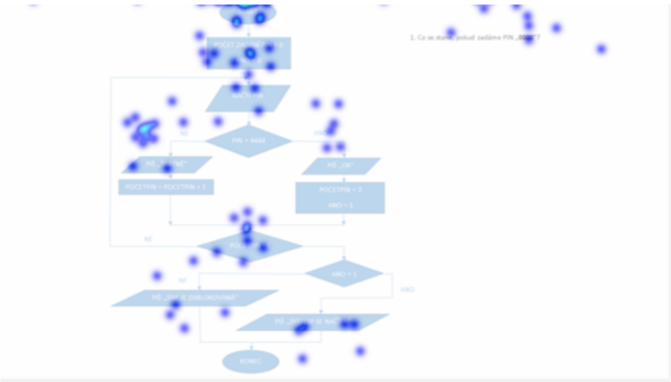


Fig. 5. Eye tracking heatmap of reading the flowchart by Respondent R3.

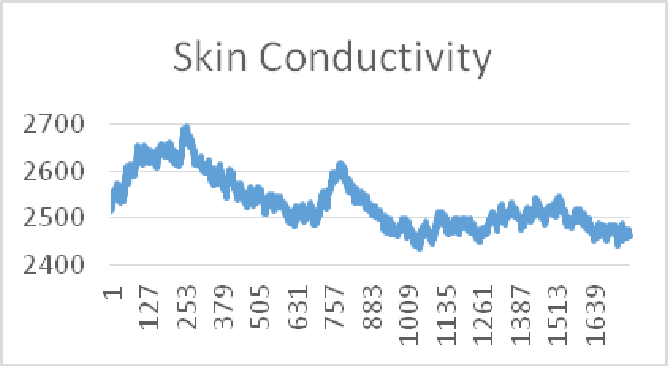


Fig. 6. GSC graph of Respondent R2. Source: Authors measuring. Key: horizontal axis – decaseconds, vertical axis – calculated quantity based on skin conductivity, in ranges of 0 to 5000.

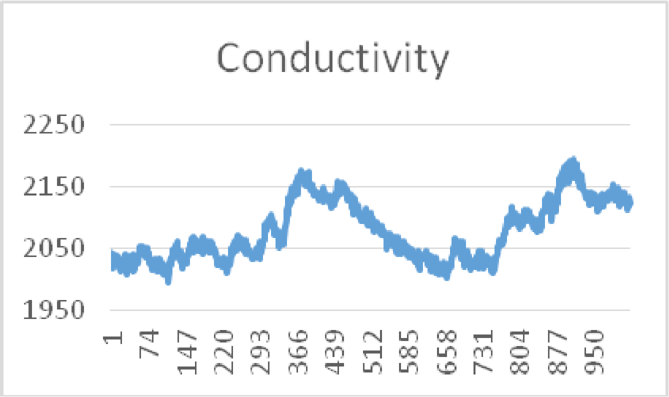


Fig. 7. GSC graph of Respondent R4. Source: Authors measuring. Key: horizontal axis – decaseconds, vertical axis – calculated quantity based on skin conductivity, in ranges of 0 to 5000.

VIII. Conclusion

Significant differences in the eye tracking records point out a different way of learning new information, which enables clustering/categorisation of these differences. Furthermore the final respondents GSC record has shown a significant relative decrease of emotion intensity, most likely caused by the fact that the respondent was a programmer. It can be generally noted that significant differences in GSC records are based on personality factors.

Although the results are statistically inconclusive (due to the small number of respondents taking part in the experiment), it can be tentatively stated that examining an individual who is solving a problem with the use of eye tracking and simultaneously noting their emotional activity (e.g. by measuring skin conductance) can objectively determine their exertion while solving the problem and at the same time highlight problematic parts of the problem-solving process. By clustering similar problem-solving progressions, which can be named serialist or holistic approaches for our purposes, we are able to select groups of students with similar thought processing approach to problem solving and provide them with procedures which are specifically adapted for these groups. This categorization is important as it enables adaptation of learning for a group of users, instead of individual users. Given the high number of students attending the introduction courses to programming and algorithmization, this is a significant factor which may lead to increasing the quality of the results of the learning process

Acknowledgment

The following grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic – GACR P103/15/06700S, Grant of SGS No. SGS 2017/134, VSB-Technical University of Ostrava. The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project ”IT4Innovations excellence in science – LQ1602”.

References

[1] M. Nakonečný, *Fundamentals of psychology*, [in Czech] Praha: Academia, 1998.

[2] J. B. Sternberg, *Cognitivne psychology*, [in Czech]. Praha: Portl, 2002.

[3] J. Mareš, *Pedagogical psychology*, [in Czech]. Praha: Portál, 2013

[4] B. Dobing, and J. Parsons, *How UML is used*. Communications of the ACM - Two decades of the language-action perspective 49(5), 109-113 (2006).

[5] G. Reggio, K. Leotta, F. Ricca, and D. Clerissi, *What are the used UML diagram constructs? A document and tool analysis study covering activity and use case diagrams*, In: Hammoudi S., Pires L., Filipe J., das Neves R. (eds.) Model-Driven Engineering and Software Development. CCIS, vol 506, pp. 66-83. Springer, Cham (2015).

[6] N. Wirth, *Algorithms + data structures = programs*. Prentice-Hall Series in Automatic Computation, Englewood Cliffs, Nj: Prentice Hall, 1976; ISBN: 0130224189

[7] ISO 5807:1985, *Information processing – Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts*. International Organization for Standardization. ISO 1985.

[8] F. B. Gilbreth, and L. M. Gilbreth, *Process charts, six steps in finding the one best way to do work*. Annual Meeting of the American Society of Mechanical Engineers. New York. 1921 s.

[9] W. Gosling, *The design of engineering systems*. New York: Wiley. 1962. 247 s.

[10] D. Moody. *The art (and science) of diagramming: communicating effectively using diagrams*, Full day tutorial at VL/HCC symposium 2012, Innsbruck

[11] Z. Dobesova, *Student reading strategies of GIS workflow diagrams*, 2nd International Conference on Humanities and Social Science Research (ICHSSR), July 29-31, 2016 Singapore, Wuhan University of Technology, China, Journal Advances in Social Science, Education and Humanities Research, Atlantis Press, ISBN 978-94-6252-235-0, ISSN 2352-5398, Vol. 70, 319-325 p., DOI: 10.2991/ichssr-16.2016.68

[12] J. Mareš, *Learning styles of pupils and students*, [in Czech]. Praha: Portál, 1998, 238 s.

[13] K. Rayner. *Eye movements in reading and information processing*. Psychological Bulletin, 1978, 85, 618-660.

[14] M. Ikeda, and S. Saida, *Span of recognition in reading*. Vision Research, 1978, 18, 83-88.

[15] M. Dragunová, R. Moro and M. Bielikova. *Visual search ability on the web*. In Proc. of IUI 2017: Int. Conf. on Intelligent User Interfaces, ACM New York, NY, pp.97-100. https://doi.org/10.1145/3030024.3038272

[16] T. Juhaniak, P. Hlaváč, R. Móro, J. Šimko, and M. Bielikova, *Pupillary response: removing screen luminosity effects for clearer implicit feedback*. In Ext. Proc. of UMAP 2016: 24th ACM Conf. on User Modeling, Adaptation and Personalisation, CEUR, Vol. 1618, 2p. http://ceur-ws.org/Vol-1618/Poster1.pdf

[17] W. Boucsein, *Electrodermal aktivity*, New York: Springer, 2012.

[18] M. Miklošíková, and M. Malčík, *Holistic and serialistic thinking as a factor influencing tex comprehension and strategy for dealing with tasks*. In Inted: 10th International Technology, Education and Development Conference. Proceedings: March 7-9, 2016, Valencia, Spain. Valencia: IATED Academy, 2016, pp. 2897 – 2902.

[19] G. Pask, *Styles and strategies of learning*. British Journal of Educational Psychology, 1976, 2, pp. 128–148

[20] J. Malach, *Programmed instruction in the context of educational theories and technologies*. In J. C. McDeermott, and B. Kožuh (eds.): Modern Approaches in Social and Educational Research. Los Angeles, USA: Antioch University, A.F.M. Krakow University, University of Primorska, 2016, pp. 65–90. ISBN 978-1-5136-1118-1.

[21] R. Móro, J. Daráž, and M. Bieliková. Visualization of Gaze Tracking Data for UX Testing on the Web. In Proc. of DataWiz 2014: Data Vizualisation Workshop (associated with Hypertext 2014). CEUR. Hypertext Extended Proceedings. Vol. 1210. 2014.

Model of education and training strategy for the management of HPC systems

Jarmila Skrinarova
Dept. of Computer Science
Faculty of Natural Sciences, UMB
Banska Bystrica, Slovakia

Adam Dudas, Eduard Vesel
Dept. of Computer Science
Faculty of Management and Informatics
University of Zilina, Zilina, Slovakia

Abstract— ***The paper is motivated by critical demand of scientists with abilities and skills in areas of mathematical modeling, simulations, big data techniques, using, administrating and management of HPC systems. We designed and realized a new model of course named "HPC system management". Its novelty lies in the content of the course, the methodology of teaching and in expected outcomes. The focus is put on job scheduling in HPC systems and management of such systems. Job scheduling is NP-complete problem. Finding a solution of a particular instance of this problem requires scientific approach. This is the reason why we choose the methodology of teaching in style of scientific project. Students created autonomous software tools and implementation of HPC system management model. The model was evaluated on the base of simulation process and last job completion time C_{max} criterion.***

Keywords— High performance computing; Management of HPC systems; HPC Strategic Research Agenda in Europe; Project education

I. Introduction

The High Performance Computing is evolving in three different directions: computing infrastructure, technological development and application development. It is necessary to adapt systems, programming paradigms and parallel algorithms. It is also necessary to manage ultra large scale parallel systems and new adaptive systems. In the future we expect that large computer systems will have ultra large scale and highly hierarchical architectures composed of computing nodes with many processors and computing cores [1], [2], [3].

To enable run of applications with increasingly larger size, there is need for new scientific methods and scalable algorithms. The emphasis is put by the published approaches in USA in context of needs for education and training specifically in HPC and Compute Intensive Science [3]. The European technology platform for high performance computing (http://www.etp4hpc.eu/), defines the HPC Strategic Research Agenda in Europe [1]. It proposes a roadmap to develop the HPC technology ecosystem and European leadership, which would be beneficial for academic and industrial users. The strategy sets objectives for various technologies required in HPC supercomputers production up to the large scale level [1]. The main challenges are specified by ETP4HPC are on the system software, system architecture and programming environment level.

HPC use model challenges are as follows: dealing with Big Data, increasing heterogeneity of data, HPC workloads in cloud computing, etc. The technical focus areas can be specified [1], [3]:

- HPC system architecture and components, energy and resiliency.
- Programming environment, system software and management.
- Balance compute, I/O and storage performance, Big Data and HPC usage models.
- Mathematics and algorithms for extreme scale HPC systems and extreme scale demonstrators.

The critical demand for new generation of educated computational scientists has been recognized in European context [4]. The International Data Corporation (IDC) report for the European Commission shows that although HPC technology is massively evolving, the effort to prepare talented students in the area is not satisfactory [5]. It is important to understand complex scientific problems and implement high quality courses to reach these objectives successfully. Our goal is to train HPC scientists with the ability and skills in mathematical modeling, simulation, computations, Big Data techniques and HPC systems using, administrating and effectively managing.

We designed and created set of courses for bachelor, master degree study and for research program [21]. This paper specifies HPC systems management course. The course is realized in master degree curriculum. A lot of universities offer various courses related to HPC. Generally, we can say that these courses are focused on models and programming techniques related to parallel and distributed computing,

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

respectively aim at creation of real or virtual systems and their administration. We offer similar courses since year 2004.

Recently, we designed and realized new model of course – HPC system management. We can describe the novelty of the course from two points of view. At first, it is in content of the course and also in methodology of course teaching. Further novelty lies in expected outcomes of the course. From content point of view there is focus put on job scheduling on HPC systems and managing models of these systems. In the course creation we draw inspiration from our experiences and scientific outputs [7, 8, 9, 10] in this area. Generally, scheduling is NP-complete problem. Modern HPC systems consist of many computing cores and they are able to run a lot of complex tasks. Finding a solution to this type of problem requires scientific approach. That is why we used methodology of teaching in style of scientific project.

The paper consists of several parts. In part Related works, there are specified and compared works that describe different approaches and implementations related to education for HPC or for creation of HPC systems. Part The model of education and training strategy for the HPC ecosystem briefly characterizes prepared courses. Next parts specify project oriented course of Management of HPC systems.

RELATED WORKS

Our previous works, which are related to management of HPC systems, are oriented on design of algorithms and models to support the scheduling tasks in grid, cloud or elastic cluster systems [7], [8], [9], [26]. This paper is also partially related to the proposal of models for parallel tasks in workflow, designed in our previous work [10], [11]. Our papers describe the e-learning courses in relation with HPC systems [6], [21].

The authors of [22] introduced their own tool (OctoShell), which shows how to help resolve daily routine problems in mastering and administering of any supercomputer center. This includes systems from stand-alone to supercomputer with many different HPC systems. The toolkit includes procedures for automation of administering and management. The authors Simons and Buell [23] are oriented on virtualizing High Performance Computing. Virtualization within HPC environments is important for bringing good performance of HPC applications in cloud computing environments. These authors discussed challenges such as workload management, scalability, interconnection, accelerators and network latency.

The authors of [24] described runtime resource management for embedded and HPC systems and presented multi-layer resource management strategy. This strategy involves optimization of applications, operating system on hardware level. At software level they exploit the Barbeque Runtime Resource Manager (Bar-bequeRTRM).

The authors Frinkle and Morris [12] are oriented on HPC cluster building. The authors described a successful approach to designing and implementing a High Performance Computing (HPC) class in Southeastern Oklahoma State University. Class is focused on creating competency in building, configuring, programming, troubleshooting, and benchmarking HPC clusters. Students built three twelve-unit independently-operating clusters. The authors of [14] described teaching high-performance computing on a high-performance cluster and demonstrated the performance aspects of 64-node Mozart cluster located at the University of Stuttgart. The authors of [15] introduced a program that combines education and research program – containing high performance computing and parallelization courses. Complex work in [17] describes the Bootable Cluster CD system running in RAM and having preconfigured environment and applications on the base of MPI, PVM, openMosix, CUDA, C and other tools and languages. Authors in [18] discuss initiatives proposed by Louisiana State University to increase knowledge and understanding of high performance computing among high school students. Czarnul [19] presented creation of the BeesyCluster environment. It is platform to make teaching and access to HPC resources easy and allows MPI, OpenMP, CUDA-level application development as well as higher level service integration using a workflow management environment. Solutions for managing HPC clusters also provide Bright Cluster Manager [20]. Bright Cluster Manager is a commercial product to deploy complete cluster over a hardware.

II. THE MODEL OF EDUCATION AND TRAINING STRATEGY FOR THE HPC ECOSYSTEM

Our model of education is motivated by critical demand of scientists with abilities and skills in areas of mathematical modeling, simulations, big data techniques, using, administrating and management of HPC systems. We have prepared in Matej Bel University (MBU), Faculty of Natural Science, in cooperation with HPCC (High Performance Computing Centre) MBU in Banska Bystrica, Slovakia, courses for undergraduate and research programs. Since 2004 we created and still are continually developing courses and modules, for reaching specified aims in context with HPC, in Applied informatics study program.

Base of compulsory and eligible courses for bachelor and master study can be found in [21]. We can briefly list courses, which are in relation with our goal: Mathematics, Algorithms and data structure, Programming, Web technologies, Computer systems architecture, Operating systems, Distributed operating systems, Computer networks, Numerical methods, MPI programming, Matrix computation, Grid and Cloud technologies, Parallel and distributed computing, Theory of algorithms, Optimization methods, Modeling and simulation. Eligible module, that is oriented on the web, grid, cloud technologies and HPC in master curriculum, consists of courses:

- Web Technologies 4.
- Big data analytics.
- Grid, cloud and HPC computing.
- Effective algorithms.
- Management of HPC systems.

For the research program there is created course for Scheduling of computer grid [6], [7], [8], [9], Load balancing in cloud computing [10],Scheduling in cloud computing [11] and Running tasks on high performance computing cluster [21]. Educational courses are obviously supported by created e-learning courses. The electronic courses are oriented to achieve quality and effectiveness in education.

Management of HPC systems course was being held previous academic year. Course consists of theoretical lectures and project creation. Theoretical part of the course is briefly introduced in part III in this paper. You can see main objectives of the course in part IV.

III. MANAGEMENT OF JOBS IN HPC SYSTEMS

In this part we briefly introduce basic model of HPC system manangement, model of job scheduling and one criterion for optimization of schedule (last job completion time C_{max}).

There is main model approach to resource management, called Workload and Resources Management System (WRMS) [26], which is used in real systems.

Model WRMS provides three types of activities:

- Managing resources, including resource management conditions.
- Managing tasks, including creating, assigning the ranks, waiting and monitoring.
- Scheduling, mapping tasks to a set of resources and the allocation of resources for certain tasks at a time.

Resource is physical component of computing system with limited capacity. Resources work with a certain velocity. Velocity of resource determines time, which is necessary for completion of a certain job on this resource [26]. In this paper, we will be considering only computational resources.

S_j denotes to start processing time of job J_j, r_j is the time availability of job j and C_j is time of job J_j completion.

We can express the main idea of scheduling in this way: A schedule means assignment of jobs to recourses for a certain time interval so that no two jobs are executed simultaneously on the same resource and resource capacity is not exceeded too. The schedule specifies for each point in time a set of tasks that are performed at that moment on a particular set of resources [26].

The most commonly used criterion for optimization of the schedule φ, that are needed to be minimized [16] is makespan. Completion time of the last job C_{max} is called the maximal time of end of job or makespan, $C_{max} = max_j\{C_1, \ldots, C_n\}$, for schedule φ is $C_{max}(\varphi) = max_j\{C_j\}$.

IV. DESIGN OF THE EDUCATIONAL OBJECTIVES AND METHODOLOGY OF THE PROJECT

Main objective of the course is to understand terms, principles, methods, algorithms, and to create a model of managing of jobs in HPC system. The model uses dynamicaly changing data from real systems. It is necessary to evaluate the model on the base of criterion in process of simulation. The solved problem is complex and it is characterized as a NP-complete problem. Subsequently, we can set up key general objectives of Management HPC systems course:

- Analysis of job scheduling on HPC systems focused on physical computational resources (see part III).
- Analysis and extraction of data from real systems.
- Creation and implementation of suitable simulation environment.
- Model creation and process of simulation of the job scheduling.
- Model evaluation on the base of specified criterion.

In summary, it is necessary to create set of software scripts and programs on the basis of analysis of computing resources and computing jobs from real extensive computing environment and its valuation by the certain criterion.

Design of methodology and tasks specification of the project in steps:

- Creation of software tools for reading characteristic of computational resources from Nordu grid monitor [13]. Extraction of data usable for creating the model.
- Create software tools for extraction of real data from characteristics of users' jobs on high performance cluster in HPCC MBU in Banská Bystrica. Extraction of the data that are usable for creating the model.
- Creation and implementation of simulator GridSim [25].
- Model creation on the base of extracted and suitable selected data. Implementation of scheduling algorithm.
- Process of simulation of the system.
- Model evaluation on the base of the last job completion time C_{max}.

The course is designed as a team project, individual tasks are assigned to students.

V. PROCESS AND OUTCOMES OF THE PROJECT

Since one of the absolvents of the course is a coauthor of the paper, this part is written from participant's point of view. We describe tasks, which were solved to accomplish the goal of the project.

A. Analysis and extraction of data from real systems

Data related to computing resources were extracted from monitoring system of internet service *Nordugrid monitor* (see http://www.nordugrid.org/monitor/) [13]. The aim of the *NorduGrid Collaboration* is to deliver a robust, scalable, portable and fully-featured solution for a global computational and data Grid system, which manages clusters of twelve countries.

The monitor publishes following information: name, number of processors (overall and occupied) and number of jobs (active and waiting) running on grid or locally, for every involved cluster. *Nordugrid* (March 2017) is managing 82 multicomputer and multiprocessor computing resources, owning 357 135 physical processors together. For illustration: there is 52 389 grid jobs running, 447 188 local jobs running, 18 119 waiting grid jobs and 54 109 wainting local jobs.

For management purposes we needed to extract big amount of data such that dynamic changes in number of arriving jobs can be seen. We created a software script, capable to read data from web page of *NorduGrid monitor*. We stored the data sets in CSV format.

The process is described as follows:

- Download of HTML document, containing all of the links to computing resources. We can find link to document containing description of certain computing resources in this document.
- We processed the documents containing the following data: name of resource, type of operating system, architecture and number of processors (on one node, overall number and occupied), grid interface, diskspace (overall, available), capacity of RAM memory, number of active and waiting jobs.
- We stored data to a CSV file. In order to model dynamic behaviour of system, we added timer to the script. The timer allowed us to store data in files every hour. Script run from 9 March 2017 to 23 April 2017. We created 1080 files of data in this period.
- We created a script which transforming the data from CSV files to format suitable for use in *GridSim simulator*.

B. Implementation of simulation environment GridSim and simulation of grid system

The GridSim implementation for purposes of modeling of job scheduling in system with physical computing resources and jobs consists of multiple steps:

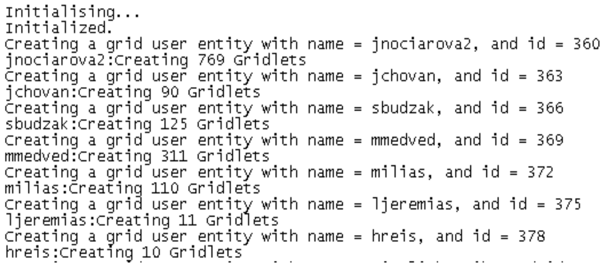
- We created *maven* project with the use of CSV file reader software. Reading data from CSV files is eqivalent to extracting computing resources available in *NorduGrid monitor*. In the establishing of *maven* project with dependencies on local jar files, there was need of correction of pom.xml file (see Figure 1) in order to work correctly with libraries and their classes.

```
      <dependency>
        <groupId>gridsim</groupId>
        <artifactId>fpv.gridsim</artifactId>
        <version>1.0</version>
        <scope>system</scope>

<systemPath>${project.basedir}/jars/gridsim.jar</
systemPath>
      </dependency>
      <dependency>
        <groupId>csv-reader</groupId>
        <artifactId>fpv.csvreader</artifactId>
        <version>1.0</version>
        <scope>system</scope>

<systemPath>${project.basedir}/jars/csv_reader-
1.0-SNAPSHOT.jar</systemPath>
          </dependency>
```

Fig. 1. Correction of *pom.xml* file

- We implemented and initialised GridSim simulator. Libraries, we imported to project as form of dependecies, are located in project of program in container archives (.jar). We created the class initializing a simulation of grid network with setup based on some of data. In the constructor of the class we added values of computing resources. The costructor initialize the network itself.
- We loaded computing resources, users and user job data to GridSim. For creation of objects describing computing resources, we used objects of *ARGridResource* type – we were using these objects for correct initialisation of number of available processors in computing machine, name of computing resource, type of architecture and operating system. Jobs are defined by ID of job, number of necessary processors, overall runtime of job (this is time defined as time from jobs availability in system to completion of job), and runtime of job (in which there is processor assigned to job). Every job is assigned to so called Gridlet. It was necessary to read list of jobs for every user in system – GridletList.
- We chose a scheduling algorithm.
- After successful loading of data of computing resources and jobs of users, simulation can be run. The simulation is started calling the method *startGridSimulation()* from GridSim class. All information about scheduling of certain jobs on prepared computing resources are stored in Gridlet objects containing jobs. We exported the results to separate files and computed last job completion time, C_{max}.

Basic output from the project is a text file consiting of thousands of lines of listing from used simulator. First part is oriented on initializing and creating grid user entities (see figure 2).

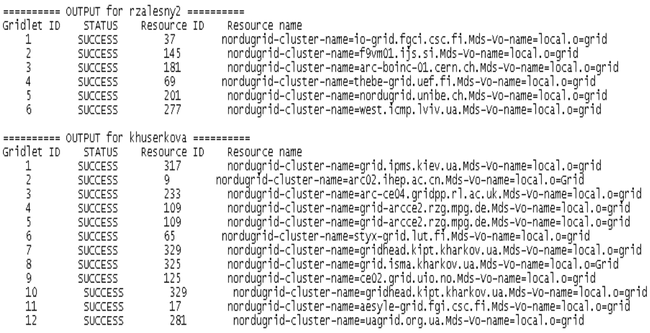


```
Initialising...
Initialized.
Creating a grid user entity with name = jnociarova2, and id = 360
jnociarova2:Creating 769 Gridlets
Creating a grid user entity with name = jchovan, and id = 363
jchovan:Creating 90 Gridlets
Creating a grid user entity with name = sbudzak, and id = 366
sbudzak:Creating 125 Gridlets
Creating a grid user entity with name = mmedved, and id = 369
mmedved:Creating 311 Gridlets
Creating a grid user entity with name = milias, and id = 372
milias:Creating 110 Gridlets
Creating a grid user entity with name = ljeremias, and id = 375
ljeremias:Creating 11 Gridlets
Creating a grid user entity with name = hreis, and id = 378
hreis:Creating 10 Gridlets
```

Fig. 2. Process of initializing and sample of creations grid user entities

Second output of the simulation is list of received gridlets (list of jobs for every user). For example number of gridlets for first three users was 769, 311 and 2636.

Simulation was finished by receiving message: „*Simulation completed. Elapsed time of simulation :: 4219 ms*.“ Sample of allocation of user gridlets to resources is illustrated in Figure 3. This allocation was created on the base of job scheduling.



```
========== OUTPUT for rzalesny2 ==========
Gridlet ID    STATUS    Resource ID    Resource name
    1        SUCCESS       37        nordugrid-cluster-name=io-grid.fgci.csc.fi.Mds-Vo-name=local.o=grid
    2        SUCCESS       145        nordugrid-cluster-name=f9vm01.ijs.si.Mds-Vo-name=local.o=grid
    3        SUCCESS       181        nordugrid-cluster-name=arc-boinc-01.cern.ch.Mds-Vo-name=local.o=grid
    4        SUCCESS       69        nordugrid-cluster-name=thebe-grid.uef.fi.Mds-Vo-name=local.o=grid
    5        SUCCESS       201        nordugrid-cluster-name=nordugrid.unibe.ch.Mds-Vo-name=local.o=grid
    6        SUCCESS       277        nordugrid-cluster-name=west.icmp.lviv.ua.Mds-Vo-name=local.o=grid

========== OUTPUT for khuserkova ==========
Gridlet ID    STATUS    Resource ID    Resource name
    1        SUCCESS       317        nordugrid-cluster-name=grid.ipms.kiev.ua.Mds-Vo-name=local.o=grid
    2        SUCCESS       9        nordugrid-cluster-name=arc02.ihep.ac.cn.Mds-Vo-name=local.o=Grid
    3        SUCCESS       233        nordugrid-cluster-name=arc-ce04.gridpp.rl.ac.uk.Mds-Vo-name=local.o=grid
    4        SUCCESS       109        nordugrid-cluster-name=grid-arcce2.rzg.mpg.de.Mds-Vo-name=local.o=grid
    5        SUCCESS       109        nordugrid-cluster-name=grid-arcce2.rzg.mpg.de.Mds-Vo-name=local.o=grid
    6        SUCCESS       65        nordugrid-cluster-name=styx-grid.lut.fi.Mds-Vo-name=local.o=grid
    7        SUCCESS       329        nordugrid-cluster-name=gridhead.kipt.kharkov.ua.Mds-Vo-name=local.o=grid
    8        SUCCESS       325        nordugrid-cluster-name=grid.isma.kharkov.ua.Mds-Vo-name=local.o=Grid
    9        SUCCESS       125        nordugrid-cluster-name=ce02.grid.uio.no.Mds-Vo-name=local.o=grid
    10        SUCCESS       329        nordugrid-cluster-name=gridhead.kipt.kharkov.ua.Mds-Vo-name=local.o=grid
    11        SUCCESS       17        nordugrid-cluster-name=aesyle-grid.fgi.csc.fi.Mds-Vo-name=local.o=grid
    12        SUCCESS       281        nordugrid-cluster-name=uagrid.org.ua.Mds-Vo-name=local.o=grid
```

Fig. 3. Part of result of simulation

VI. EVALUATION OF THE COURSE

Usage of data from real environment helped students to better understanding of system functionality and methods of measurement in HPC systems. Solving of beneficial problems, necessary for practice and saving time and electric energy acts as motivation and activating stimulus for students. Students perceive positively the connection of theory and practice and reaching of complex solution as an outcome of the project. Practical experiences in creation of project may help students to get better work positions.

The main objective of the practical part of course was to understand terms, principles, methods, algorithms and to create a model of managing jobs in HPC, which uses dynamicaly changing data from real systems. Specific objectives of this course were to create the set of software scripts and programs on the basis of analysis of computing resources and computing jobs from real extensive computing environment, and its valuation by the certain criteria.

This objectives have been successfully and completely fulfilled. As you can see from A6 in Figure 4, there was small problem in implementation optimization of scheduling algorithm. Students first analyzed system data of resources in *NorduGrid* and data about user jobs from HPC MBU cluster. They created scripts, programs and implemented their own autonomous software tools and implemented simulator GridSim. By implementation of all parts of the management process and schedulling jobs to particular resources students gained deep understanding of terms and principles. As we can see from output of the project, process of simulation is equivalent to model creation of managing jobs in HPC, which uses dynamicaly changing data from real systems (which was part of the main goal).

Seven students completed Management of HPC systems course successfully. Students were graded on the basis of theoretical and practical part of the work. Practical part is displayed in Figure 4 by activities A1 – A7 for every student separetely. Evaluation of theoretical part of the work is displayed by T1 – T7. Theoretical part of the examination consists of three questions from the set of eleven theoretical theses, which are not shown here for the lack of space. Grade was weighted as 50% theoretical part and 50% practical part. Overall evaluation of course is: two students reached the grade A, three B, one C and one student got the grade D. Overall average grade of study group is 1.5 – B.

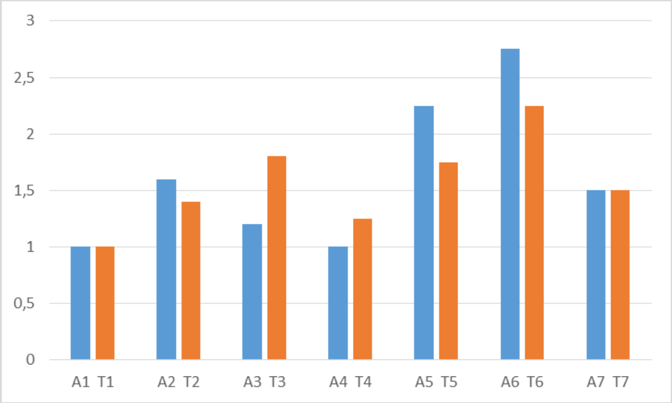


Fig. 4. Evaluation of HPC course

Activities allocated to students: A1 - Creation of software tools for reading characteristic of computational resources from Nordugrid monitor, A2 – CSV reader tool, A3 – Creation of software tools for extraction of real data from characteristics of users jobs on high performance cluster in HPCC MBU in Banská Bystrica, A4 - Creation and implementation of simulator GridSim, A5 - Implementation of scheduling algorithm, A6 – Creation and optimization of scheduling algorithm, A7 - Process of simulation of the system.

From the point of view of the course life cycle, feedback should be incorporated in the next run of the course. In the next run, a teacher should reserve more time to work with scheduling algorithms. This means that lecturer should set control points in such way, that there is enough time for implementation of tools applicable in the end of the task. This is an inspiration for building a new project in which some of the software tools created in the current project can be re-used. This will provide a scope for implementing existing or design and creation new scheduling algorithms for HPC systems. We can draw inspiration from scientific tasks, which were solved in the past [7], [8], [9], [26], or from new research tasks from the area of virtual machine scheduling in cloud systems, or management of systems with regard to minimization of electric energy use.

The study program was evaluated by 11 students on the basis of 16 questions. Out of these 16 question only three were substantial for our work: Relevance of courses in study program (graded 1,59), logical succession and continuity of courses (graded 1,45) and actuality of content of study program (graded 1,50). In overall evaluation was study program graded by grade from these possibilities: 1 – very high level of quality, 1,5 - high level of quality, 2 - medium level of quality, 2,5 - low level of quality , 3 – very level of quality. Study program was graded by students with the grade of 1,5 - high level of quality.

ACKNOWLEDGEMENTS

Part of the computing was performed in the High Performance Computing Center of the Matej Bel University in Banska Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

The research was partially supported by the grant of The Ministry of Education, Science, Research and Sport of the Slovak Republic, VEGA 1/0487/17.

REFERENCES

[1] ETP4HPC, ETP4HPC Strategic Research Agenda Achieving HPC leadership inEurope, 2013, http://www.etp4hpc.eu/wp-content/uploads/2013/06/ETP4HPC book singlePage.pdf.

[2] T. Van der Pyl, The European HPC strategy and actions in Horizon2020, DGConnect, European Commission Feb. 2014., 2014,http://www.apre.it/media/181410/van der pyl.pdf.

[3] M.-R. Sancho, "ESC Best Practices in Professional Training and teaching for the HPC Ecosystem". Journal of Computional Science 14, 2016, pp.74-77.

[4] EC COM, Communication from the Commission to the European Parliament,The Council, The European Economic and Social Committee and the Committeeof the Regions: High-Performance Computing: Europe's place in a Global Race, European Commission, Brussels, 15.2.2012, COM (2012) 45 final, 2012.

[5] E. IDC Joseph, et al., "High Performance Computing in the EU: Progress on theImplementation of the European HPC Strategy Report", ISBN978-92-79-49475-8., 2015.[9] PATC OMB, PRACE Advanced Training Centres HPC Curriculum 2014,PRACE IP3 April 2014, 2013.

[6] J. Skrinarova, M. Krnáč, "E-learning course for scheduling of computer grid", International Conference Interactive Collaborative Learning ICL 2011. Wien : International Associaton of Online Engineering, 2011. IEEE Computer Society, - ISBN 978-1-4577-1746-8. - pp. 352-356.

[7] J. Skrinarova., M. Krnac, "Particle Swarm Optimization Model for Grid Scheduling," in Second International Conference on Computer Modelling and Simulation, CSSIM 2011. Brno, Czech Republic. pp. 146-153.

[8] J. Skrinarova, L. Huraj, V. Siladi, "A neural tree model for classification of computing grid resources using PSO tasks scheduling," in Neural networks world. 2013. ISSN 1210-0552 .

[9] J. Skrinarova., M. Krnac, "Particle Swarm Optimization for Grid Scheduling:, in Eleventh International Conference on Informatics, 2011, Rožňava. pp. 153-158.

[10] J. Skrinarova., M. Povinsky, "GPGPU based job scheduling simulator for hybrid high-performance computing systems". IEEE 13th International Scientific Conference on Informatics, 2015, Location: Poprad, SLOVAKIA, 2015, pp. 269-274.

[11] J. Skrinarova, "Implementation and evaluation of scheduling algorithm based on PSO HC for elastic cluster criteria," in Central European Journal of computer science. 4 (3) 2014. pp. 191-201.

[12] K. Frinkle, M.Morris, "Developing a Hands-On Course Around Building and Testing HPC Clusters", ICCS 2015 International Conference On Computational Science, pp. 1907–1916.

[13] NorduGrid: htttp://www.nordugrid.org.

[14] M.Bernreuther, M. Brenk, H.Bungartz, R. Mundani, I. Muntean. "Teaching high-performance computing on a high-performance cluster". In Sunderam et al. [14], pp. 1–9.

[15] M. Berzins, R.M. Kirby, C. R. Johnson. "Integrating teaching and research in hpc: Experiences and opportunities". In Sunderam et al. [14], pp. 36–43.

[16] Paluch, S., Pesko, S. *Quantitave methods in logistics* (In Slovak) Kvantitatívne metódy v logistike. ŽU Žilina. 2006. ISBN-80-8070-636-0, pp. 185.

[17] A. Gibbon, D. Joiner, H. Peck, S.Thompson. "Teaching high performance computing to undergraduate faculty and undergraduate students". In Proceedings of the 2010 TeraGrid Conference, TG '10, pp. 7:1–7:7, New York, NY, USA, 2010. ACM.

[18] S. R. Brandt, Chirag Dekate, Phillip LeBlanc, Thomas Sterling. "Beowulf bootcamp: Teaching local high schools about hpc". In Proceedings of the 2010 TeraGrid Conference, TG '10, pp. 4:1–4:7, New York, NY, USA, 2010. ACM.

[19] P. Czarnul. "Teaching High Performance Computing using BeesyCluster and Relevant Usage Statistics", ICCS 2014. 14th International Conference on Computational Science, Procedia Computer Science, Volume 29, 2014, pp. 1458–1467.

[20] Bright Cluster Manager: http://www.brightcomputing.com/product-offerings/bright-cluster-manager-for-hpc

[21] J. Skrinarova, E. Vesel, "Model of education and training strategy for the high performance computing", 14th IEEE international conference on Emerging eLearning technologies and applications, IEEE, 2016. ISBN 978-1-5090-4699-7. - CD-ROM, pp. 315-320 (2016).

[22] D. Nikitenko, V. Voevodin, S. Zhumatiy. 2016. "Resolving frontier problems of mastering large-scale supercomputer complexes". In Proceedings of the ACM International Conference on Computing Frontiers (CF '16). ACM, New York, NY, USA, 349-352. DOI: https://doi.org/10.1145/2903150.2903481.

[23] J. Simons, J. Buell. 2010. "Virtualizing high performance computing". SIGOPS Oper. Syst. Rev. 44, 4 (December 2010), pp. 136-145. DOI=http://dx.doi.org/10.1145/1899928.1899946.

[24] W. Fornaciari, G. Pozzi, F. Reghenzani, A. Marchese, M. Belluschi. 2016. "Runtime resource management for embedded and HPC systems". PARMA-DITAM '16. ACM, New York, NY, USA, pp. 31-36. DOI=http://dx.doi.org/10.1145/2872421.2893173.

[25] Buyya, R., Murshed, M.M : GridSim: a toolkit for the modelling and simulation of distributed resource management and scheduling for Grid computing. *Concurrency and Computation: Practice and Experience.* 2002. pp. 1175-1220

[26] Skrinarova, J.: *Elastic cluster*. (In Slovak) Elastický klaster. Banská Bystrica:2017, Belianum ISBN 978-80-557-0642-9

Assembling Behavioural Characteristics of Malicious Software

Jana Šťastná and Martin Tomášek
Department of Computers and Informatics
Technical University of Košice, Košice, Slovakia
{jana.stastna, martin.tomasek}@tuke.sk

***Abstract*—In this paper we present our approach to assembling characteristics of malware, which may be useful for another researchers, in case they face limitations of hardware and software resources during their work. Our research aims at exploring malware characteristics on high level of abstraction and finding how much abstract can the characteristics be to bring results. To pursue our goals, we needed to analyse great amount of malware samples. Unfortunately, we faced obstacles which did not allow us to effectively perform malware analysis for obtaining data we required. As a result, we came up with different approach for assembling data form malware analysis by employing online resources. The paper describes realization of our approach, which allowed us to assemble large amount of malware behavioural characteristics and utilise them in our research.**

I. Introduction

At early malware's history, a program that could detect malicious file would simply look for exact fragment of code, sequence of bytes, which was found to occur in instances of that malware type. To complicate analysis of new malware samples and thwart extraction of the incriminating malicious sequences of code, malware writers started to protect their code with obscuring methods, which evolved into sophisticated techniques of analysis and detection evasion.

Malware research is nowadays undergoing a change of strategy. While syntactic representation of malicious features is still forming a core for many anti-virus scanners, malware researchers claim that in order to catch up with new threats novel forms of malware representation and characteristics tied with malware behaviour, not malware syntactic structure, need to be aimed at. Research made in accord with this direction does not instantly lead to improvements in anti-virus products, nor is it the goal of our work. What needs to be achieved first is finding incriminating malware characteristics which are hard to obscure, and determining level of abstraction which would be suitable for their representation, thus to be employable in the future, perhaps in new kind of detection signatures.

In our work in general we focus on researching levels of abstraction on which malware characteristics and behaviour can be extracted. There is a great difference between describing malware on the level of sequences of operational code, and level of function calls to system API. Our goal is to explore high level of abstraction concerning malware characteristics, find out how much abstract can the characteristics be and investigate limits of such approach. However, for the research we needed to analyse great amount of malware samples. Unfortunately, dynamic analysis requires that every sample is executed for at least several minutes and static analysis requires coping with obfuscation, packing and other anti-reversing measures. Since researching malware characteristics would take very long time in that manner, we addressed gathering data about malicious samples with different approach, which forms the core part of this paper.

This paper is presenting our, rather untraditional, approach to collecting characteristics of malware. First, we explain our motivation behind seeking a different approach for accumulation of data about malware, by presenting problems associated with malware analysis (Sec. II). Next we present how we implemented our approach in a form of software tool (Sec. III), which collects information about malware from online resources. A preliminary delineation of our approach has been given in our previous paper [1] and in the following text we offer an extensive description of our work. The application of our software tool in malware research is described here only briefly (Sec. III-C), since other of our papers, referred also further in the text, deal with that in full extent.

II. Motivation and Goals

Malware is crafted in such way that analysing it requires carefully prepared analytic environment, a lot of various analytic tools, and most of all, plenty of time to employ various analytic techniques, to uncover incriminating malicious features and behaviour. These are serious problems if you try to research malware in a university environment, with limited hardware, software and financial resources, and within a small research team.

A. Malware Samples Accumulation

Resources of malware samples should be clearly specified. It is important to be able to replicate the experiment, as also explained in book of Josiah Dykstra [2], so it is good to document methods of malware accumulation or online resources used.

A weak spot of malware research is inclusion of harmless files in experiments. We do not satisfy with files solely from clean Windows installation as a representative set of harmless files, as presented e.g. in work of Islam et al. [3]. Since users normally work not only with applications provided as part of operating system installation, but also with a lot of software,

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

clean programs, provided from "the outside" of the OS, we believe that those should also be part of harmless samples set, if such set is included in the experiment. As Rossow et al. discuss in their work [4], it is not always appropriate to include harmless programs as research samples, but if the experiment requires it, a wide variety of harmless samples is useful.

Amount of malware samples used in research experiments is also in the area of interest. An example how it should be implemented is work of Jacob et al. [5], in which their system for malware similarity comparison and packer detection has been evaluated with a sample set containing *795 thousand* samples. In our work we aimed to follow their example.

B. Analytic Environment Detection

Researching malware behaviours usually requires controlled infection of a computer system which is reserved for analytic purposes. It is necessary to secure the environment so that malware cannot exploit the stage of dynamic analysis, to e.g. employ network connection for spreading into another systems. On the other hand, we need to guide the execution so that malware performs its malicious activity to the full extent. As a result, a conjunction of strict security measures and execution conditions leading to full activity of observed sample needs to be arranged. There are several options for realization of these terms, described e.g. by Egele et al. [6], however, they have weaknesses.

The first option is to allocate a physical system as analytic environment. A notable advantage is that such environment will resemble a system of causal computer user. With high probability, malware will manifest its full malicious potential in such configuration. A crucial requirement is isolation of the analytic system from the rest of the network, so that malware will not spread. However, there is a risk that security measures will be circumvented. The best solution would be to completely physically separate the system from the network, but this could lead to malware limiting its behaviour, due to unavailable network connection.

Another drawback of a physical system as analytic environment is that malware may exhibit different actions under different operating systems, so observing this behaviour would require several physical machines with different operating systems installed on them, or one physical system with dual-boot configuration. Unluckily, both solutions are time-demanding to manage. After infecting the system with malicious sample and analysing it, the analytic environment needs to be "cleaned", i.e. brought back to the initial state as before malware infection. This often entails reinstalling the operating system and configuring the environment from the scratch. Even if this can be mitigated by preparing the installation as an image of the operating system together with all software analytic equipment beforehand, but the installation only will still require a lot of time. In order to analyse large amount of malware samples a different approach must be applied.

The second option available is configuring a virtual system or several virtual systems on one physical host system. The advantage is that each of them may run different operating system. Several virtual systems connected into a network will allow analysing interaction of samples under such conditions. Managing virtual systems is also much simpler. An image of the system with all configurations and required analytic software can be easily prepared, so the virtual system can be quickly restored into initial, clean state. Possibility to create snapshots of the system can be used for analytic purposes, as in work of Wagener, State and Dulaunoy [7]. Recording state of the system after infecting it with malware allows to make comparisons with the initial system configuration. A disadvantage of this approach is requirement of large hard disk space, since system images are stored in form of relatively large files. Especially, analysing numerous malware samples may get impractical because of that. Also, running several virtual systems on one physical host system is demanding on amount of operating memory.

The most serious problem with virtual system as analytic environment is that malicious programs are able to detect such system. Song, Royal and Lee demonstrated in their experiments how can malware efficiently circumvent analysis [8]. There are slight differences between physical and virtual systems which malware can recognise, and thus modify its execution path accordingly, to show no malicious activity. Lindorfer, Kolbitsch, and Comparetti [9] listed analytic environment detection techniques classified according to distinguishing features in hardware, environment, software and network settings. An ideal solution would be to develop a fully transparent virtual analytic environment, i.e. an environment unrecognisable from physical host system. That, however, is not feasible easily. Balzarotti et al. discuss problems accompanying the struggle for analysis transparency [10] and although they presented an approach to identify malware which is able to detect analytic environments, their technique is not suitable for malware analysis on large scale.

C. Static and Dynamic Analysis Evasion

Techniques of static analysis are commonly employed due to their accessibility, variability, or availability of means for automation. They allow researchers to trace flow of data or various execution paths implemented in program's code, or observe similarities of two program samples on the level of sequences of operational code. Unfortunately, when static analysis grew in popularity, malware writers fought back and increased defences, hiding crucial segments of code from analysts' sight by packing and obfuscation techniques.

Packing can be simply described as a combination of encryption and compression which thwarts analysability of code. Packed samples can be analysed statically only after unpacking. Analysts can overcome simple packers by static unpacking - without executing the sample. Sophisticated malware, however, is modified by custom-made packers and getting around them requires executing the program, so that packer reverses modifications made to the code, one instruction by another. Running every analysed program, even just for few minutes, is a time-consuming task. As a result, benefits of static analysis are lost.

Fortunately, according to Jacob et al. [5], similarity even of packed programs is measurable without executing them. While their technique doesn't allow to observe behaviour of packed programs, only detect their similarity, it is a great benefit, since syntactically equivalent samples will be equivalent in behaviour.

Obfuscation is another method of code modification on syntactic level. It introduces various, even useless, instructions into program's code in order to perplex it and increase amount of time required for analysis above acceptable level. A variety of forms of obfuscation are delineated in a survey of You and Yim [11].

Success of static analytic techniques is endangered also by so-called *fileless malware*. It leaves no traces in form of files during execution, resides in memory and may hide as a part of a legitimate system process [12]. These tricks enable malware to avoid detection systems which employ static analysis and scanning of the filesystem. Although fileless malware appears rarely, it portrays an unpleasant situation when there is simply no file to analyse.

As an answer to static analysis countermeasures, dynamic analysis gained popularity. By observing program execution it allows to capture actual behaviour performed by analysed program, so false execution paths purposed to confuse reverse engineers can be easily disregarded. A frequently used analytic method is function call monitoring which allows to record which specific functions from Windows API, Windows Native API, or system calls were called by analysed sample. It may seem that dynamic analysis defeats malware stealth, however, Egele et al. mention [6] that malware samples exist, which are able to avoid the system call interface and so circumvent dynamic analysis.

D. Summary of Issues Addressed in the Paper and Goals

Malware analysis is a time-consuming activity which requires repetition of numerous tasks. Analysing a sample with static analysis, executing a sample in a controlled environment for dynamic analysis, while capturing as much activity of a sample as possible, assembling the results into a comprehensive analytic report and cleaning or rebuilding the analytic environment for analysis of another sample. This is mere a concise enumeration of tasks required for malware research.

The experiments should be performed with as many samples as possible for obtaining relevant results and reaching objective conclusions. Accumulating thousands of samples, be it with help of honeypots, web crawlers or public malware databases, delays the actual malware research.

Each of the problems presented in sections II-A–II-C may form extensive research areas and one cannot address them all on his or her own. Therefore, after considering our hardware and software resources and estimating the time needed for our work, we searched for resources readily available for simplifying collection of data about malware.

Services for malware analysis provided online seemed like a solution to our problem. In addition to analysing provided files, databases with reports from such inspection are often available too, via web interface. We investigated several such services, e.g. Totalhash, Malwr, VirusTotal, Hybrid-analysis, with emphasis on amount of already processed malware samples, diversity, and accessibility of data they provided. To explore characteristics and behaviour of malicious programs, we needed to access large amount of reports from analysis. Obtaining them by hand would be time-inefficient, therefore we needed to build a software tool, which would access analytic service of our choice and collect desired data.

The objective of our work was to implement a software tool which would be able to do the following:

- access reports of malware analysis on selected online service,
- collect data from the reports,
- store the data into database in a unified format, so the data could be analysed in our research.

III. Solution for Assembling Malware Characteristics

Considering issues which appear when one tries to assemble malware characteristics for analysis (Sec. II), we addressed the problems by implementing a software tool, presented in the following paragraphs. We show that collecting data about large amount of malware samples is feasible even in a small team of malware researchers with limited hardware and software resources and in a relatively short time, and we present how such goal can be achieved, to help another malware researchers gain data for their experiments.

Reports form malware analysis produced by different online analytic services contain different types of data and structure. Unifying them would bring complications into implementation, so we resorted to selecting one of the services for current implementation, while making the tool extendible later with another analytic services.

The service of our choice was Totalhash[1]. It allows to obtain analytic reports in two ways: through API access or through a web browser. In case of API it's easy to employ Totalhash's services programatically, however, in case of unpaid access one can request only 300 reports per month, which is too few for research purposes. More reports are available through API only with paid subscription, which also didn't meet our requirements. However, we were able to take advantage of the structure of service's web pages, so we could access analytic reports with the web browser and the task was simplified and automatized by employing a web crawler.

The tool for collecting data from Totalhash was implemented as a command-line application. We called it *TH Downloader* and it formed a start of our research project *ViLMA*.

A. Structure of TH Downloader

TH Downloader consists of three main modules called *Site*, *Link* and *Report*. Each of the modules can run independently from the others, but this is a recommended order in which

[1] Available online at: www.totalhash.com

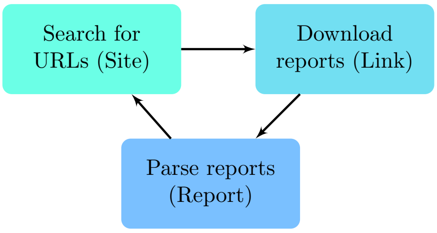


Fig. 1. Three main modules of *TH Downloader* and cycle of their operation.

those modules should operate (Fig. 1). Because the analytic service employed for data collecting may limit the amount of analytic reports available per day, and the processing speed of individual modules of the tool may change according to system performance and network connection, the processing limit may be set for each of the modules.

1) Module Site: It is responsible for collecting web links (URL addresses) to individual reports which contain malware analysis results. Based on the structure of Totalhash's web site, it has to parse its *Search* web page which contains a list of URL addresses of reports and store them for further processing (Fig. 2). The list is paginated, with each page containing 20 entries. When the end of the page is reached, the module will switch to the next page. Before the module ends its execution, it stores a position from the list on which it ended, so when the module is executed again, it will continue from the latest processed entry.

2) Module Link: It works with the list of links (URL addresses) collected by the previous module (Site). The module requests an HTML document from the link, containing the analytic report and stores it as a file to the hard drive (Fig. 3). Since in case of Totalhash the URL address of a report is made of hash code which identifies the analysed sample, downloaded reports have clear and comprehensible labels. The reports are saved for archiving purposes. In case of changing the strategy for processing data from reports, it is not necessary to download them again.

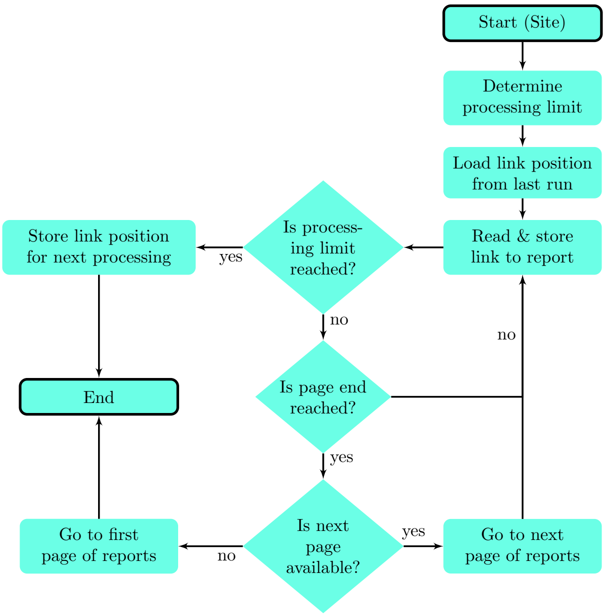

Start (Site)
Determine processing limit
Load link position from last run
Read & store link to report
Is processing limit reached?
yes
Store link position for next processing
End
no
Is page end reached?
no
yes
Is next page available?
yes
Go to next page of reports
no
Go to first page of reports

Fig. 2. Flow diagram of module Site of *TH Downloader*.

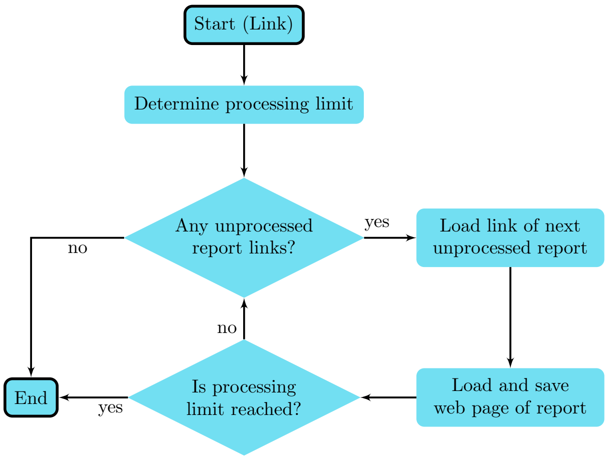

Start (Link)
Determine processing limit
Any unprocessed report links?
yes
Load link of next unprocessed report
no
End
no
Is processing limit reached?
yes
Load and save web page of report

Fig. 3. Flow diagram of module Link of *TH Downloader*.

3) Module Report: It processes HTML documents containing reports from malware analysis, collected by module Link (Fig. 4). This module may have numerous alternative implementations, based on what data from reports are preferred for current research purposes. Since reports contained in HTML documents may be very extensive or very short and their contents are not unified, it was necessary to decide on fundamental information which should be targeted for parsing by this module. More about data contained in reports is explained in Sec. III-B. Output of the parsing process is an object containing information according to predefined database structure, which can vary for different alternatives of the module.

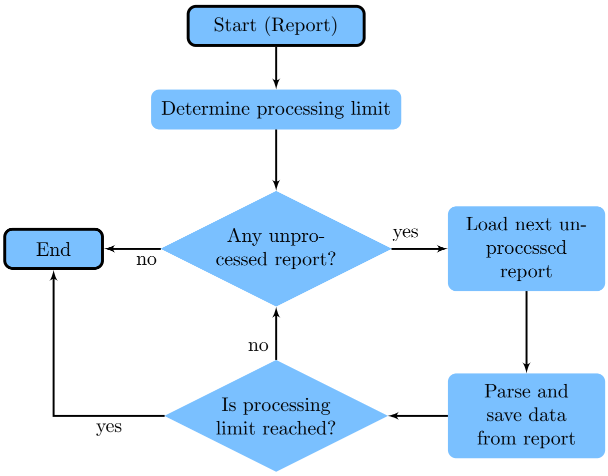

Start (Report)
Determine processing limit
Any unprocessed report?
yes
Load next unprocessed report
no
End
no
Is processing limit reached?
yes
Parse and save data from report

Fig. 4. Flow diagram of module Report of *TH Downloader*.

B. Data accumulated by TH Downloader

Reports of malware analysis produced by Totalhash have not all the same contents. Their structure, however, is usually almost the same: First there is date and time of analysis, and hash codes as unique identifiers of the file (MD5 and SHA1). Then there are three main sections which contents may vary depending on results of analysis:

Static Details describe type of analysed file, names, hashes and sizes of program's sections (in case of PE executable for OS Windows), packers that were detected, and importantly, results of scanning by almost 30 anti-virus (AV) engines. If an AV engine detects no threat, a message "no virus" is shown. In the other case, a name of detection signature is shown, which informs how is this instance of malware identified by given AV. An important note here is that signature names are not unified among AV products, which is a great, long known problem for malware research, as noted eg. by Canzanese, Kam and Mancoridis [13] or Mohaisen and Alrawi [14].

Runtime Details describe activity of analysed sample during its execution, i.e. when analysed dynamically. The section contains details concerning executed processes, modifications of files, registry entries and mutexes, DNS requests or additional sub-processes created by the sample.

Network Details allow a closer look at network activity of analysed sample. The section describes e.g. TCP data flows, DNS and HTTP requests.

Additional data from analysis are available at the end of a report, in text fields *Raw Pcap* - containing unprocessed captured network traffic, and *Strings* - containing file names, IP addresses, application messages, or many other readable information extracted from the executable file.

We observed significant differences in contents of analytic reports from Totalhash. While some reports contained minimum of information and some data were missing, e.g. detected packers, network activity, or strings found in the executable, others were in contrast vast. To cope with that we selected a set of information which was sufficient for our experimentation and at the same time reachable by parsing module of *TH Downloader*.

The database which stores data parsed by module Report (see Sec. III-A) contains two tables - *antivirus* and *report*. Table *antivirus* stores data from static analysis and results from scanning the sample by AV engines which are currently available in Totalhash. Table *report* stores output from dynamic analysis and describes activity of analysed sample concerning file actions, processes, registry entries, or network activity. Additional data extracted from reports are text fields Strings and Raw Pcap. As these can grow very long and often exceed limits of standard database data types, they are not stored in the database. If found in the report, they are saved as text files to the hard drive and their location in the file system is stored into database table *report*.

C. Application of TH Downloader in Our Research

Research of malware high-level characteristics assembled thanks to *TH Downloader* is fully presented in our other paper [15]. Next follows a shortened description of how these data were used in our work.

Each report obtained by our tool delineates program's activity which was performed during dynamic analysis. All the data are too complex to use directly, therefore we transformed behavioural information of analysed samples into numeric values, which represent amounts of actions performed in selected behavioural categories. Our intention was to obtain behaviour representation on a very high level, which, to our knowledge, has not been experimented with yet. Of course, large amount of information about malware samples is neglected this way, but our goal was to experimentally verify whether transforming information from malware analysis into such high-level representation may lead to some helpful conclusions.

Based on types of information which occurred in analytic reports from Totalhash, we selected 12 categories of behaviour for our experiments: *file creation, file deletion, mutex creation, process creation, service creation, service starting, registry entries, DNS, Winsock DNS, HTTP get, HTTP post, TCP flows*. These actions concern malware samples which targeted OS Windows, therefore other malware samples (e.g. for mobile OS Android) were neglected for the time being.

Data of each malware sample conforming 12 categories were abstracted into numeric values and formed a 12-tuple of numbers, describing sample's behaviour on very high level of abstraction. Analysis of these data was performed with database queries in development environment *DataGrip*[2]. Analytic reports were first clustered by results of anti-virus detection, i.e. by name of malware signature. Each cluster of entries comprised data about syntactically different malware samples but detectable with the same signature. Next, in each cluster we analysed number of actions executed by samples (values from 12-tuple among entries) and figured their minimal, maximal and average value. Data analysis revealed that such values may characterise malware samples conforming the same signature, e.g. if minimal and maximal number of file creations among samples were equal, it indicated that file creation is a characteristic for samples of that malware signature. In several cases, all of numbers from 12-tuple characterised samples of malware signature, e.g. we found malware signature from which every sample created exactly 10 new files, deleted 2 files, created 4 mutex entries, created 1 new process, operated with 3 registry entries, performed 3 DNS and 3 Winsock DNS requests and used 4 TCP data flows. The rest of behaviour categories from 12-tuple showed value 0. These findings led us to defining behavioural patterns on high level of abstraction, which definition is also given in our other paper [15].

We measured in how many cases it is possible to extract behavioural patterns from data collected by *TH Downloader*. This work is presented in other of our papers [16]. In summary, significant differences between labelling systems used by AV-engines cause problems in such measurements. While one of them recognised 11 338 distinct malware signatures among

[2] Available online at: https://www.jetbrains.com/datagrip/

samples which reports we analysed, another found only 1 656 malware signatures. This labelling inconsistency is a persisting obstacle in malware research, as mentioned in Sec. III-B.

We compared pattern extractability with respect to data about malware signatures provided by various AV engines. Despite differences, we could state that for all AV engines on average 10.72% of malware samples could be defined by behavioural pattern, and in case of three AV engines it is more than 15% of samples.

While extractability of behavioural patterns on high level of abstraction is, according to our experiments, not very high, our results show that there is a notable portion of all malware samples that potentially does not require more than observing amount of actions executed for detection or identification.

IV. Conclusion and Future Work

In relation to issues discussed in Sec. II-A, employing online malware analytic services eliminates the problem of malware samples accumulation and troublesome handling. The analytic service archives all the samples that have been analysed and analytic reports produced by the analysis, so data applied in our experiments can be found and used for replication by other researchers. Considering that malware samples are identified in analytic reports by unique hash code (SHA1 and MD5), they can be traced in another analytic services as well, or obtained from web pages which provide malware samples for research purposes (e.g. malwr.com).

Developers behind the Totalhash service are well aware of issues concerning building a malware analysis environment and techniques which malware creators use to detect and circumvent it (Sec. II-B). Their strategy for outsmarting malware was to combine strengths of numerous malware analytic environments, e.g. Cuckoo Sandbox, Anubis, ThreatExpert, Malwr, CWSandbox, which they mention in their blog [3]. Totalhash employs a custom analytic environment which is not distributed to the public. This measure of secrecy is reasonable, considering tendency of malware writers to develop tactics for circumventing accessible analytic environments. Consequently, Totalhash maintains success in triggering malicious behaviour in sophisticated malware.

Employing online resources with massive amount of data allowed us to implement research experiments in a short time and verify our hypothesis about level of data abstraction for malware research. We already work on upgrading *TH Downloader* to increase its efficiency with parallel processing, and additional analytic service will be employed as data resource - VirusTotal[4]. To enhance research applications of our tool we will employ search and analytic engine Elasticsearch[5], which will help to work with long text entries, such as readable strings extracted from executable files. Number of analysed characteristics will increase and more detailed information about program's behaviour will be extracted from analytic reports, e.g. activity performed by each of processes that a program executed, names and paths to modified files, keys and values of modified registry entries. Another data about malware behaviour are assembled. Currently, more than *300 thousand* new analytic reports are at hand to evaluate our behavioural patterns for detection purposes.

Acknowledgment

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-15-0055 and by project KEGA no. 079TUKE-4/2017.

References

[1] P. Hlinka, M. Tomášek, and J. Šťastná, "Collecting significant information from results of malicious software analysis," *Electrical Engineering and Informatics 7*, pp. 103–108, 2016.

[2] J. Dykstra, in *Essential Cybersecurity Science*. O'Reilly Media, Inc., 2016, ch. Cybersecurity Experimentation and Test Environments.

[3] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 646 – 656, 2013.

[4] C. Rossow, C. J. Dietrich, C. Grier, C. Kreibich, V. Paxson, N. Pohlmann, H. Bos, and M. v. Steen, "Prudent practices for designing malware experiments: Status quo and outlook," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 65–79.

[5] G. Jacob, P. Comparetti, M. Neugschwandtner, C. Kruegel, and G. Vigna, "A static, packer-agnostic filter to detect similar malware samples," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. Lecture Notes in Computer Science, U. Flegel, E. Markatos, and W. Robertson, Eds. Springer Berlin Heidelberg, 2013, vol. 7591, pp. 102–122.

[6] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surv.*, vol. 44, no. 2, pp. 6:1–6:42, 2012.

[7] G. Wagener, R. State, and A. Dulaunoy, "Malware behaviour analysis," *Journal in Computer Virology*, vol. 4, no. 4, pp. 279–287, 2008.

[8] C. Song, P. Royal, and W. Lee, "Impeding automated malware analysis with environment-sensitive malware," in *Proceedings of the 7th USENIX Conference on Hot Topics in Security*, ser. HotSec'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 4–4. [Online]. Available: http://dl.acm.org/citation.cfm?id=2372387.2372391

[9] M. Lindorfer, C. Kolbitsch, and P. Milani Comparetti, *Detecting Environment-Sensitive Malware*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 338–357.

[10] D. Balzarotti, M. Cova, C. Karlberger, and G. Vigna, "Efficient detection of split personalities in malware." in *NDSS Symposium 2010*, 2010. [Online]. Available: https://www.ndss-symposium.org/ndss2010/efficient-detection-split-personalities-malware/

[11] I. You and K. Yim, "Malware obfuscation techniques: A brief survey," in *2010 International Conference on Broadband, Wireless Computing, Communication and Applications*. IEEE, Nov 2010, pp. 297–300.

[12] L. Zeltser, "The history of fileless malware - looking beyond the buzzword," 2017. [Online]. Available: https://zeltser.com/fileless-malware-beyond-buzzword/

[13] R. Canzanese, M. Kam, and S. Mancoridis, "Toward an automatic, online behavioral malware classification system," in *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems*, Sept 2013, pp. 111–120.

[14] A. Mohaisen and O. Alrawi, *AV-Meter: An Evaluation of Antivirus Scans and Labels*. Cham: Springer International Publishing, 2014, pp. 112–131.

[15] J. Šťastná and M. Tomášek, *Characterising Malicious Software with High-Level Behavioural Patterns*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2017, vol. 10139, pp. 473–484.

[16] J. Šťastná and M. Tomášek, "High-level malware behavioural patterns: Extractability evaluation," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems (to be published)*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 11. IEEE, 2017, pp. 569–572.

[3] Available online at: https://totalhash.cymru.com/blog/page/4/

[4] Available online at: https://www.virustotal.com/

[5] Available online at: https://www.elastic.co/products/elasticsearch

Using software support to implement enterprise controlling

Katarína Teplická
Institute of earth resources
Technical university, Faculty BERG
Košice, Slovakia
katarina.teplicka@tuke.sk

Soňa Hurná
Department of Air transport management
Technical university, Faculty of Aeronautics
Košice, Slovakia
sona.hurna@tuke.sk

***Abstract*— improving business success means improving business processes and using software support for process management. The offer of software products for enterprise controlling implementation is extensive. The most well-known software products are Monet ABC, SAP, Gradient, SAS, EIS Dominant. In the paper we point out the benefits of software support in business process management. Methodologically, we will use economic analyzes and statistical methods to track changes in processes. Controlling is instrument for continually evaluating of processes.**

***Index Terms*—controlling, software support, improvement, efficiency, business**

I. Introduction

Improving business success means improving business processes and using process-oriented support tools to achieve key factors such as efficiency, economy and productivity.

Refäuter [2] characterized controlling as a specific form of work with information, and its role is not to direct real business processes, but the entire enterprise through information about real business processes. Dried defined controlling as a management approach that allows for the setting of real goals that, once approved by management, become binding targets. Pupil defined controlling as a business management system based on a comprehensive information and organizational link between the planning and control process, which aims to improve business results. Freiberg explains functional controlling as a business management subsystem focused on the planning and control process and on its information support. According to R. Mann [2], controlling is profit-oriented management and supervision. Controlling is profit management. E. Mayer [2] characterizes controlling as a kind of corporate governance. It is a profitable management. Gurcik [2] defines controlling as an integrative management tool that supports corporate decision-making and management. According to Ľ. Kalafutova [2] is a controlling continuous process focused on economic control of a business through economic information. P. Horvath [2] defines controlling as a management subsystem focused on the planning and control process and on its coordination and **information support [7]**.

II. Significance of controlling

The essence of controlling is to achieve economic transparency; transparency in business activities, processes, planning, calculating, budgeting, process decisions, orders, performance, customers, products [8]. All businesses are looking for an optimal solution for implementing controlling. For the implementation of controlling, the use of Microsoft Office, MS Access, MS Excel and a specialized controlling system (CIS), presents the minimum financial demands. Currently, there are a number of companies offering controlling information systems, SAP, ABRA G2, CUNT-KUD, EPICOR iSCALA, EVIS / 400, GIST Controlling, HELIOS GREEN, HERKULES SOFTIP, MonetABC [10].

III. Analysis of barriers of controlling implementation

Controlling is part of the management information and economic systems. The application of controlling requires a change in the company's management system, a change in the organizational structure of the company, the creation of a controller's position, a controlling body. With controlling implementation are connected all barriers in Slovak companies [5].

TABLE I. Barriers of controlling implementation in Slovak companies

Area	Barriers
Management	Problems in changing mentality and worker habits with regard to increased activity and effective behavior. Leaders' orientation towards short-term goals, which is not rational in terms of strategic goals. Conflict of the process of evaluating the results, in the insufficient differentiation of factors acting on the efficiency of independent economic centers.
Personal	Small and overloaded controlling staff, occupied by a number of tasks (3 to 5 people). Lack of staff capable of performing the role of controller and responsible for the

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

	implementation and operation of controlling. Difficulties in selecting the staff suitable for the function of the controller. Relatively low level of training for the controlling workers. Lack of knowledge about the use of controlling tools.
Organization structure	Problems associated with changing the organizational structure of the business. Inappropriate inclusion of the controller in the organizational structure of the enterprise and improper definition of its competencies. Management decentralization issues related to increased decision-making autonomy. Limited time to adapt verified practices and management concepts to specific business conditions.
Evidence and analyses	Weaknesses in the detailed and systematic control of total costs, in a large enterprise are the verification of the regularity of order management in written form very administratively demanding and difficult. Fixed and variable costs are considered to be unambiguously variable and unambiguously variable. Problems with a rigorous restructuring of the material and value planning system as well as the information system. Difficulties in the operational planning of substantive tasks in service activities. Absence of standards and indicators of the degree of wear of the production facility. Difficulties in terms of term tasks, accounting for budgeting of material and value objectives. Inappropriate way to convert inputs to outputs. An inadequate reporting system, which does not draw any specific conclusions, does not get the lead of the company.
Information	Current state of the enterprise equipment by computer technology and lack of special programs supporting the implementation of controlling. Lack of open new networking computer network modules for business needs, while registration is often done manually.
Finance	Lack of funding for the employment of qualified business professionals in small businesses. Lack of funds to buy the necessary computing. Few opportunities for adequate financial incentives to meet the objectives set by middle management.

In this part we analyze software support for controlling.

SAP has the largest share in the Slovak market in the provision of software products. The Controlling (CO) module has a significant position within the SAP system. It represents an important tool for strategic planning and management, allowing it to constantly monitor and manage costs, revenues, results, resources, deadlines and deviations [9]. The CO module is integrated with other system modules. The CO model allows you to perform various controlling processes: Plan the economic result and costs - create cost centers, create a sales plan, create orders, plan calculations, create profit center plans up to the overall plan of the organization, debit real costs and revenues - possibility to analyze the causes of their occurrence, to compare the centers with each other and to monitor the internal business outcome, Closing of the accounting period - includes the state of completion and analysis of production costs, the calculation of the overheads of overheads. The CO model includes a whole set of output reports that are the basis for analyzing and evaluating planned and actual data and helping to generate forecasts and proposals for measures for the future.

CUNT-KUD – this system belongs to the Czech company, which represents the management tool for management to decide on the production and business strategy of the company. It allows you to reduce costs, increase productivity and control the management of the centers. By introducing data into the controlling system and then comparing them with actual accounting and operational-technical evidence, deviations will be made on the basis of which the realities of the objectives will be assessed and the direction of the development will be determined.

EPICOR iSCALA

The iSCALA Enterprise Information System is the product of an international information company, which is also represented in Slovakia and one of the most important providers of enterprise information systems. Controlling software support created under the Kontroling module is the primary analytical and reporting tool of the iSCALA enterprise information system for organizing information sets, improving accessibility, and clarifying information. The specified parameters can be integrated into the iSCALA system or act as a standalone application. The Kontroling Module is a versatile and powerful tool for accessing information and provides several unique features: A transparent application with a graphically displayed data organization provides easy usage, Organizing tree information menus in a structured tree provides users with quick and easy access to information, Can also be used as inputs in other assemblies. In this way, the transfer of results for the further processing of the information can take place without the information leaving the security of the software. Each user has only his / her data available without another user entering them. The outputs can be created in formats: XML, Microsoft Excel, Crystal Reports graphical output, viewing table, eventual analysis and data change.

EVIS / 400 a modular system in which the Controlling Module can perform all planned and actually overhead and unit costs for individual cost centers and workplaces and from them to production and commercial contracts. After every monthly ending period, when all cost information is available then calculations are made. From the accounting module and the production planning and control module, the necessary data is automatically downloaded. Calculated data can be checked during and after the computation process. This controlling module is particularly applicable to manufacturing plants.

GIST controlling - software product is operated in the Slovak and Czech Republics in almost all business sectors. GIST Controlling is from the Czech company GIST it is a controlling and superstructure management system, which is composed of individual functional areas focused on a particular area. These areas include: marketing and sales, profitability, purchasing, inventory, production, investment, deviation analysis, payment calendar, and reporting.

ONIX - enterprise Information System Developed by KROS, a. s. It is enterprise software for middle sales and purchasing management. The goal of Onix is to simplify the life of the business, to streamline and improve internal processes and save time and money. The system is designed for business and engagement companies that help not only handle orders, invoice and inventory, but also evaluate profitability in clear and comprehensible graphs. The advantage of this software is more efficient company management, clear purchasing and sales analysis, as well as a perfect overview of the contract, a sophisticated warehouse management, software tailored to the needs of the professionals, and a simple and user-friendly environment.

IV. Using of controlling software support

Software support is very effective and important instrument for processes in the firm and their evaluation. Controlling is instrument for very economic and effective evidence and for creation database in the firm with all important information [6]. We present one very important instrument for improvement of managerial decision in the firm. It is controlling software. Fig. 1 describes report of assets of the firm and information of plan and reality in one year. Than contents all parts of assets for example building and costs.

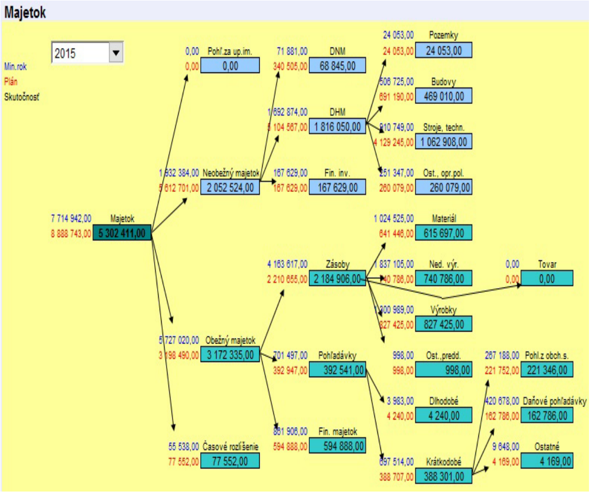
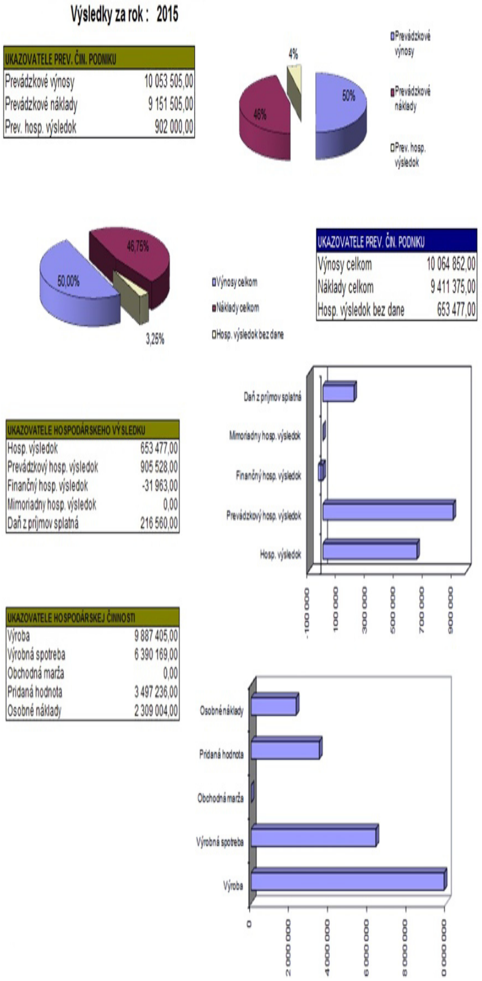


Fig. 1. Analyses of assets in the firm by controlling instrument.

Complex economic analyses of production in the firm represent figure 2. This is one part of controlling software in this firm. In this report are information about revenues, costs, profit, tax, production, added value of business and other. Each report is processed by graphical and calculated form. In this reports are very important information for management and for performance management of the firm.



Výsledky za rok : 2015

UKAZOVATELE PREV. ČIN. PODNIKU	
Prevádzkové výnosy	10 053 505,00
Prevádzkové náklady	9 151 505,00
Prev. hosp. výsledok	902 000,00

UKAZOVATELE PREV. ČIN. PODNIKU	
Výnosy celkom	10 064 852,00
Náklady celkom	9 411 375,00
Hosp. výsledok bez dane	653 477,00

UKAZOVATELE HOSPODÁRSKEHO VÝSLEDKU	
Hosp. výsledok	653 477,00
Prevádzkový hosp. výsledok	905 528,00
Finančný hosp. výsledok	-31 963,00
Mimoriadny hosp. výsledok	0,00
Daň z príjmov splatná	216 560,00

UKAZOVATELE HOSPODÁRSKEJ ČINNOSTI	
Výroba	9 887 405,00
Výrobná spotreba	6 390 169,00
Obchodná marža	0,00
Pridaná hodnota	3 497 236,00
Osobné náklady	2 309 004,00

Fig. 2. Reports of business processes by controlling approach. .

Controlling is a management tool that can keep businesses in an efficient way and avoid existing problems that are often a matter of "survival" for businesses [4]. Controlling emphasizes to business managers how to achieve more effective management and business goals. Controlling expresses the complex function of economic management in enterprises [3] and we can understand all the activities related to the entrepreneurial activity of the enterprise. The reason for poor implementation of controlling in enterprises is insufficiently trained controlling personnel, not built [1]. The management information system and its connection to the economic and accounting system of the company, the lack of personnel competent to carry out the control function, the poor organizational structure of the company, insufficient registration system, shortcomings in the interconnection of economic analyzes, unprocessed management accounting, unfinished planning and budgeting in the enterprise, building controlling, undefined strategic goals of the business. These reasons can also lead companies to avoid controlling.

TABLE II. BENEFITS OF CONTROLLING IMPLEMENTATION IN SLOVAK COMPANIES

Area	Benefits
Planning, motivation,	Improvement of the structure of plans, extension of plans containing external parameters, assets and liabilities, financial flows and selected indicator values. Improve the accuracy and quality of planning by identifying them by senior staff of those departments that require increased attention. Improving the effectiveness of the enterprise's management activity by deepening the knowledge of the weaknesses and strengths of the business, its bottlenecks. Increasing the motivation of executives at lower levels of management by linking tasks to the motivation system of their assessment.
Efficiency	Improving control activity. Possibility of monitoring, measurement of indicators. Higher quality of processed analyzes.
Strategic Goal	Greater credibility of the data. Precise formulation of tasks, business objectives. Decentralization of decision making. Improving the flow of information. Improved knowledge of middle-level managers.
Evidence and information system	Enhancement of evidence. Extension of the record area. Acceleration of information flow. Improving business communication across centers and between centers and economic departments.
Budgeting	Unification of cost budget. Implementation of calculation of contributions to be paid. Adaptation of systems for the differentiation of economic centers.

V. CONCLUSION

The use of business process controlling is very effective and necessary in the long run. The programming languages are very important instrument for creation of controlling software and controlling applications [13]. Structural operational semantics is most popular semantic method for software engineers and this method is used by creation of software for controlling [14]. Minimizing process costs can bring benefits to eliminate the risk and preference of the economic effect [11, 12]. Business processes cannot be managed without changes and improvements, which can ultimately affect the price of products, demand and meet customer requirements, increased revenue from product sales, lower total costs, and profit growth. In addition to financial benefits, it is also necessary to perceive another aspect of this approach, namely to improve the information transfer system at individual management levels, to provide feedback in synergy with the customer, to eliminate ineffective activities in the chain of business processes [2]. Through controlling, key business processes can be successfully managed and enable the enterprise as a whole to build a concept of continual continuous improvement.

ACKNOWLEDGMENT

This contribution was supported by project VEGA 1/0741/16 Innovation controlling of industry companies for maintenance and improvement of competitiveness and KEGA 002 TUKE - 4/2017 Innovative didactic methods in education at university.

References

[1] A. Foltínová, Ž. Kalafutová, “Controlling in the firm, “, Bratislava, Elita, 1998.

[2] A.R. Eschenbach, “Controlling,“ Praha, Aspi Publishing, 2004.

[3] D. Baran, “Application of controlling in praxis, “Bratislava, 2006.

[4] I. Váryová, A. Látečková, “Controlling and financial accounting, “Nitra, 2011.

[5] J. Kádárová, M. Durkáčová, “Risk management in industrial companies, “ in IEEE – SAMI 2012, Slovakia, January 2012, p. 415-419.

[6] J. Nesterak, V. Bobáková, “Controlling – new system of management, “Bratislava, Ekonóm, 2003.

[7] K. Čuchranová, “Cost management of the firm as basic instrument for economic efficiency”, International conference - Metallurgy Junior, Košice, 2001, p. 109 –117.

[8] M. Potkány, A. Šatanová, “Controlling – modern tool of company control, “in Economic journal, Bratislava, 2004, p. 148-165.

[9] M. Potkány, M. Hitka, “Controlling conception of integral management with motivation programm, in Management in theory and praxis, vol. 1., issue 3, 2005, p. 42-48.

[10] R. Rajnoha, “Planning and Controlling,” in Journal of theory and praxis of management, vol.1, Zvolen, 2011.

[11] Š. Kassay, “Firm of international class,“ Nové Zámky, Strateg, 2001.

[12] T. Petřík, “Economical and financial management of the firm,“ Praha, Grada Publishing, 2005.

[13] W. Steingartner, V. Novitzká, “A new approach to semantics of procedures in categorical terms”, in IEEE – Informatics 2015, Poprad, November 2015, p. 252-257.

[14] W. Steingartner, V. Novitzká, “Categorical model of structural operational semantics for imperative language”, in Journal of Information and Organizational Sciences, Zagreb, vol. 40, issue 2, 2016, pp. 203-219.

Detection of shoe sole features using DNN

Michal Vagač, Michal Povinský and Miroslav Melicherčík
Faculty of Sciences, Matej Bel University, Banská Bystrica, Slovakia
Email: {michal.vagac,michal.povinsky,miroslav.melichercik}@umb.sk

***Abstract*—Shoe prints belong among the most common types of evidence at crime scenes. To determine the brand or manufacturer of a shoe, it must be matched with known collection of shoe print samples, or with shoe sole pictures. Comparing the shoe print to the shoe sole picture is difficult task, because of different nature of data. Therefore it is more convenient to compare shoe sole features instead of raw image data. In this paper we focus on detection of the shoe sole features in picture of shoe sole using Deep Neural Network.**

I. Introduction

Shoe prints are some of the most common evidence left at crime scenes. After identification of an object (a shoe), which made the print, the object can be used as an evidence. Therefore determination of class of the object (in this case it is manufacturer and brand of the shoe) helps to successful resolution of the case. Nowadays, the process of determination of shoe brand is mostly manual. Therefore it is very demanding, time consuming and costly. Automatization of this process would lead to more efficient processing of available evidence.

To enable automatic identification of given shoe print, a reference database of prints is required. However, building of such database is considerably time (and financially) demanding. To create one database sample, a shoe must be physically available, then the shoe's sole is printed on special paper, which is finally scanned to digital form. To build such a big database in this way is at least very laborious (and costly) process.

In our research, we aim to create (and subsequently to query) the database of photographs of shoe soles. The task of creating the database is easily achieved considering photographs of shoes (and also shoe soles) available on the Internet. However, querying the database with image of shoe print is challenging – there is a need of comparison of different types of data (Fig. 1). We need to define such features, that are consistent in both types of images – photography and print.

Such features can be different patterns and shapes that form the shoe sole. They are visible at both types of images. In image data, they are described by edges, which are surrounding them. Edge detection is a fundamental tool in the area of feature detection in computer vision. However, applying standard edge detectors to picture of shoe sole brings two problems: 1) incomplete or weakly visible edges are not detected, 2) the detectors react to small details, which are not important (are even disturbing). A reason for these results lies in fact, that the ordinary edge detectors does not consider overall context, but reflects only the nearest surrounding of processed pixel. For this reason, we have implemented edge detection using convolutional neural network. By training the network with proper data, we can get detection of those edges, which meet our requirements for good shoe sole features.

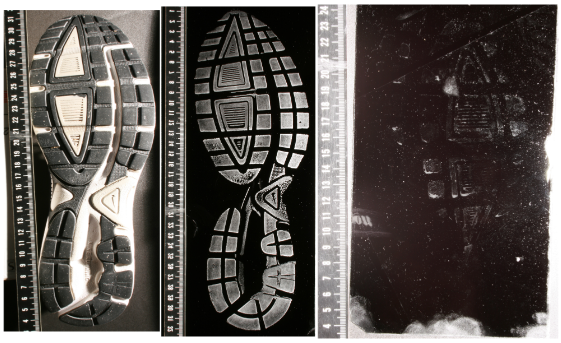


Fig. 1. Types of used images from left to right: photography of a shoe sole, a shoe print made in laboratory on special gelatin lift, and a shoe print secured at crime scene

II. Related Work

Several papers has been published on the topic of footwear recognition. Main focus of these works is on recognition part of the process. In the most cases authors describe their methods, which allows to find given foot or tire print in existing database of prints. Common process is to build query image by modifying an image from database (using blur, rotation or crop actions).

Geradts and Keijzer [1] employed a coding scheme based on series of geometric shapes automatically generated from footwear prints. At first, the image is thresholded, then morphologic operations are applied to suppress noise. This image becomes subject of connectedness analysis. The spatial position of these shapes is recorded. To each type of geometric shape, the two-dimensional Discrete Fourier transform (DFT) was applied and the final matching was achieved with a neural network processing the Fourier transform coefficients.

Chazal et al. [2] proposed a system for automatic sorting of the database of shoeprints based on the outsole patterns in the Fourier domain in response to a reference shoeprint. A pattern is described using power spectral density (PSD) of the print. A correlation function of the PSD coefficient from a reference database and a query image is used as a similarity metric. The footwear prints were processed globally and hence noise in the images could have hindered the quality of useful encoded local information evident in the print.

The technique for shoe print image retrieval and classification suggested by [3] is based on extraction of local features (MSER and SIFT). MSER features are transformed to SIFT descriptors and encoded relative to a feature codebook forming histogram representations of each shoe pattern. Codebook is constructed by pooling together all the feature descriptors from the pattern database and then using k-means clustering to quan-

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

tize this data into a fixed number of bins. Each pattern is then encoded using term frequency–inverse document frequency (tf–idf) weighting.

Nibouche et al. [4] proposed a solution for matching rotated partial shoeprints. Harris points encoded with SIFT descriptors are used as features and they are matched using random sample consensus (RANSAC). The proposed solution in this paper tackles the issues of rotation and noise distortion in partial prints. The number of detected points is reduced by creating a 4-level pyramid where a detected point is only taken into account if its Laplacian response is a local maxima in 3x3x3 neighbourhood.

Dardi et al. [5] described a texture-based retrieval system for shoeprints. A Mahalanobis map is used to capture the texture and then matched using a correlation co-efficient measure. The paper focuses on geometrical structures observed as distances between pixel value patterns. The relative positions and shapes of different parts in the sole print texture region are common for a given shoe model. They divide a gray scale image area into several sub-regions or blocks, and calculate the distances between all possible block pairs.

Other paper [6] proposes clustering based on recurring outsole patterns. The clustered database is used to retrieve similar prints for a given crime scene mark. Geometric shapes like line segments, circles and ellipses are proposed as features for crime scene marks. Then, these features are structurally represented in the form of an attributed relational graph (ARG). Robust ARG matching is achieved with the introduced footwear print distance (FPD), a similarity measure for footwear prints. The proposed system is invariant to scale, translation, rotation and insensitive to noise and degradations of the prints.

III. Implementation

A. Dataset

The training data was prepared from 13 source images, obtained from the Internet. The lines were first detected using LSD (Line Segment Detector) algorithm, then cleaned up manually. The resulting lines were then rendered into image files with the same resolution as the input files. Each image was rendered at 36 different rotations, giving us 468 training image pairs. 10 of these images were randomly selected for the validation set, however the network performance was primarily evaluated manually on a separate set of images. When training, a random crop of 640x640 was selected and batch size of 1 was used. Adam optimizer was used with initial learning rate of 0.001 and learning rate was decayed by 0.98 every 10 epochs.

B. Network

The network was implemented in Torch framework [7] using Lua language. It's architecture (Fig. 2) consists of 3x3 convolution layers with ELU (Exponentional Linear Unit) activation function. The layers were plugged together using `nn.Sequential` class, as depicted in the following fragment:

```
local bw = 14
local n = nn.Sequential()
n:add(nn.SpatialConvolution(3, bw, 3, 3))
n:add(nn.ELU(nil, true))
```

No batch normalization or dropout is used. There are three nested modules, each halving the scale and doubling the number of channels. The modules use strided convolution to reduce input resolution and upsampling on the output. The output is concatenated in the channel dimension with the module's input.

```
local inner
for k,v in ipairs({bw*4, bw*2, bw}) do
  local v2 = v*2
  local seq = nn.Sequential()
  seq:add(nn.SpatialConvolution(v, v2, 3, 3,
        2, 2, 1, 1))
  seq:add(nn.ELU(nil, true))
  local w = v2
  if inner then
    seq:add(inner)
    w = w * 2
  end
  seq:add(nn.SpatialConvolution(w, v, 3, 3,
        1, 1, 1, 1))
  seq:add(nn.ELU(nil, true))
  seq:add(nn.SpatialUpSamplingBilinear(2))
  inner = nn.Concat(2)
  inner:add(nn.Identity())
  inner:add(seq)
end
n:add(inner)
```

Bilinear upsampling is used in the network. Upsampled outputs from the subnetworks are concatenated with the output of the last layer.

The numbers of layers, channels and nested modules was determined empirically to achieve acceptable output quality while keeping the computational costs low. All convolutions are padded to preserve image size. Every convolution layer except the final convolution is immediately followed by ELU activation function. The final convolution layer is followed by sigmoid activation. Its bias was initialized to -3.8, which was determined empirically to minimize oscillation of the Adam optimizer.

```
local lc = nn.SpatialConvolution(bw, 1, 5, 5,
        1, 1, 2, 2)
lc.bias:add(-3.8)
n:add(lc)
n:add(nn.Sigmoid())
```

The network is trained using binary cross-entropy loss, as the problem can be viewed as a per-pixel binary classification problem. It has 110497 parameters.

IV. Experiments and Results

When evaluating, the image was split into 192x192 tiles with 16 pixels of overlap. Empty tiles were skipped. This image was rotated by 90° and the process was repeated for every possible rotation. All resulting tiles were then combined using the maximum function.

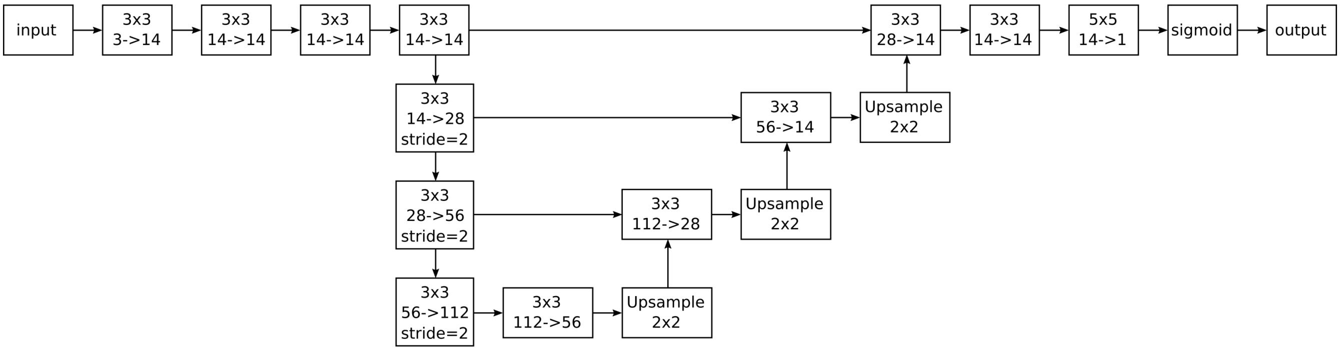


Fig. 2. Proposed network architecture

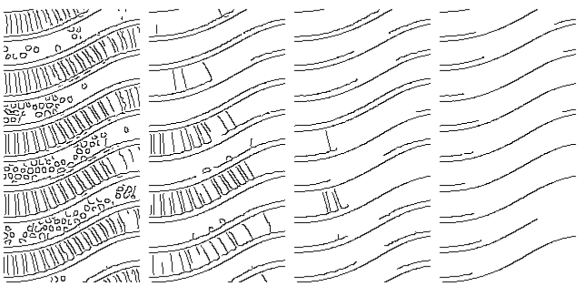


Fig. 3. Example of detection results using Canny edge detector. After parameters tuning and ignoring small edges, also main edges are affected

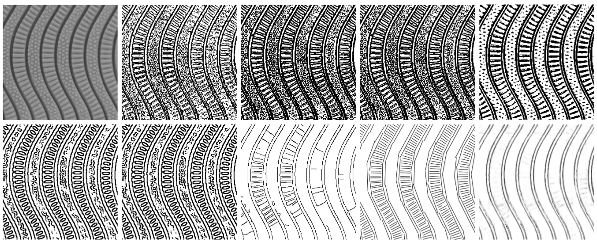


Fig. 4. First row from the left: input image, Roberts, Prewit, Sobel, Haralick; second row from the left: Marr-Hildreth, Marr-Hildreth-log, Canny, LSD, our solution

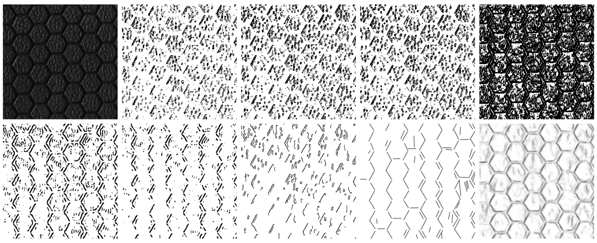


Fig. 5. First row from the left: input image, Roberts, Prewit, Sobel, Haralick; second row from the left: Marr-Hildreth, Marr-Hildreth-log, Canny, LSD, our solution

The resulting image was manually compared to images generated using standard edge detectors [8]. Figure 3 explains deficiency of standard edge detectors (in this case Canny edge detector). The aim is to remove unimportant edges, but to preserve significant ones. After parameters tuning, when unimportant edges disapear, also significant ones are affected.

Comparisons of standard edge detectors results with results obtained by our network are shown at figures 4, 5, 6 and 7. Expected results for images at figures 4 and 5 are detected edges of waves and hexagons respectively. Since the images contains slight texture, results of standard edge detectors contains considerable amount of noise.

Similar results are visible at figures 6 and 7. Detection results contains either too much noise, or are missing edges describing subjected feature. By tuning parameters it is possible to balance between these two states. However, often it is impossible to balance out to get sufficient results. The problem is even harder considering tuning parameters for a whole set of images.

V. Conclusion

In this paper we have applied a deep convolutional neural network for detection of shoe sole features. From depicted results it can be seen that the network gives better results of edge detection than standard detection techniques in terms of our requirements.

Obtained detection results can be further improved. One deficiency, which we have observed, is related to imperfection of manual preparation of training data. This can be improved by more careful cleaning of LSD detected line segments and reconsidering rules, about which edges should be marked. Further, we plan to evaluate the results using k-fold cross-validation technique to estimate accuracy. Expanding the training dataset and providing images from currently poorly detected classes (different colors, lighting, shapes) of images could improve the results. Data augmentation techniques could also be used to make the results more consistent.

Acknowledgment

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0219-12.

Computing was performed in the High Performance Computing Center of the Matej Bel University in Banska Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

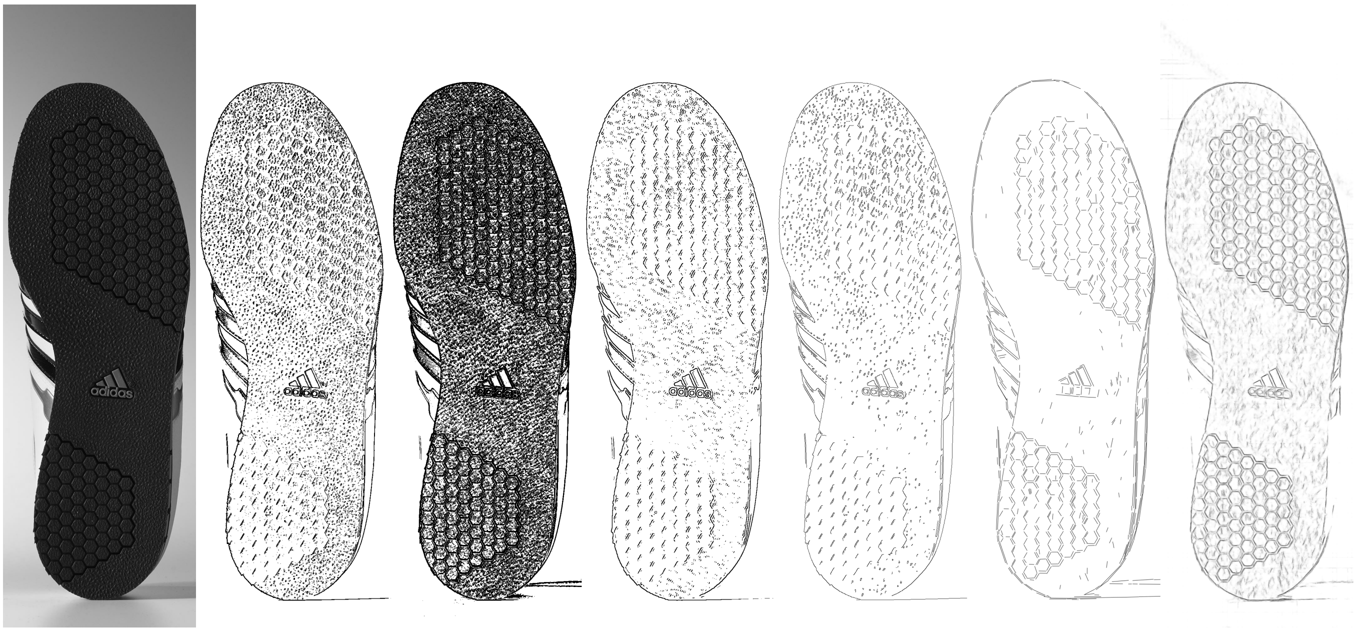


Fig. 6. From left: input image, Sobel, Haralick, Marr-Hildreth-log, Canny, LSD, our solution

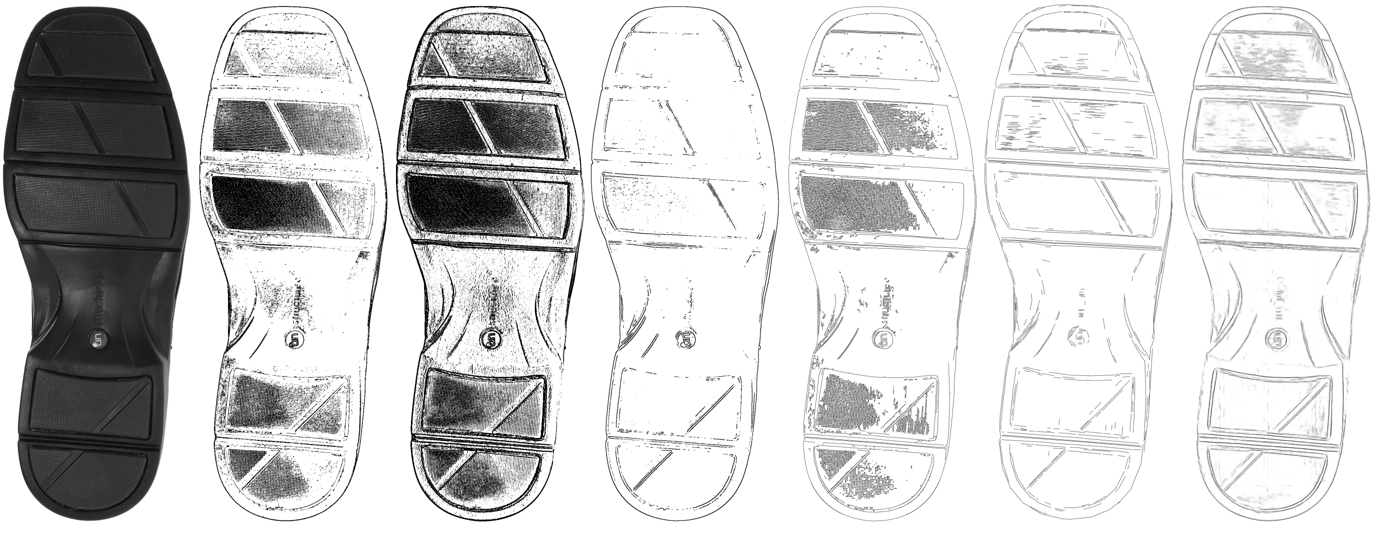


Fig. 7. From left: input image, Sobel, Haralick, Marr-Hildreth-log, Canny, LSD, our solution

REFERENCES

[1] Geradts, Z., Keijzer, J.: The image-database {REBEZO} for shoeprints with developments on automatic classification of shoe outsole designs. Forensic Science International, vol. 82, no. 1, pp. 21–31 (1996)

[2] Chazal, P., Flynn, J., Reilly, R.B.: Automated processing of shoeprint images based on the Fourier transform for use in forensic science. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 3, pp. 341–350. IEEE Computer Society Washington, DC, USA (2005)

[3] Pavlou, M., Allinson, N.: Automated Encoding of Footwear Patterns for Fast Indexing. In: Image and Vision Computing, vol. 27, no. 4, pp. 402–409. Butterworth-Heinemann, Newton, MA, USA (2009)

[4] Nibouche, O., Bouridane, A., Crookes, D., Gueham, M., Laadjel, M.: Rotation Invariant Matching of Partial Shoeprints. In: Proceedings of the 13th Machine Vision and Image Processing Conference, pp. 94–98. IEEE Computer Society, Washington, DC, USA (2009)

[5] Dardi, F., Cervelli, F., Carrato, S.: A Texture Based Shoe Retrieval System for Shoe Marks of Real Crime Scenes. In: Proceedings of the 15th International Conference on Image Analysis and Processing, vol. 5716, pp. 384–393. Springer-Verlag Berlin, Heidelberg (2009)

[6] Tang, Y., Kasiviswanathan, H., Srihari, S.N.: An efficient clustering-based retrieval framework for real crime scene footwear marks. In: Int. J. Granular Computing, Rough Sets and Intelligent Systems, vol. 2, no. 4, pp. 327–360 (2012)

[7] Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A Matlab-like Environment for Machine Learning. In: BigLearn, NIPS Workshop (2011)

[8] Spontón, H., Cardelino, J.: A Review of Classic Edge Detectors. Image Processing On Line, 5 (2015), pp. 90–123.

Linear Logic Operators in Transparent Intensional Logic

Liberios Vokorokos
Department of Computers
and Informatics
Technical University of Košice
Letná 9
042 00 Košice, Slovakia
e-mail: liberios.vokorokos@tuke.sk

Zuzana Bilanová
Department of Computers
and Informatics
Technical University of Košice
Letná 9
042 00 Košice, Slovakia
e-mail: zuzana.bilanova@tuke.sk

Daniel Mihályi
Department of Computers
and Informatics
Technical University of Košice
Letná 9
042 00 Košice, Slovakia
e-mail: daniel.mihalyi@tuke.sk

***Abstract*—In this paper, we would like to describe two logical systems – Tichý's transparent intensional logic and Girard's linear logic. Transparent intensional logic is a method of logical analysis of natural language, which represents expressive system, independent of chosen source language. Our research vision is to connect this method with linear logic. In future research, this connection would allow us to apply Ludics theory principles (which is the extension of the linear logic) to natural language sentences, which puts them in a logical time and space.**

I. Introduction

Logic is the science of correct argumentation and reasoning. For the correct reasoning it is necessary to understand the expressions of the language and know how to determine their sense. In computer science, various logics are traditionally applied to artificial formal languages. However, mainly in the field of artificial intelligence, there are requirements for the processing of natural language by computers, not only on the syntactic but also on the semantic level.

Linear logic (LL) [1], [5] in computer science is not only used in linear logic programming [16], modeling of Petri nets [18] and IDS behavior [17], [22], etc. LL may also describe the dialogues in natural language [10], [20] and capture their sense. For this purpose, the Ludics theory [11], [21] is used as a LL extension with time-spatial proof calculus. The fact that LL can be used as an instrument of logic analysis of natural language (LANL) operating on the semantic level of language, links it to the Transparent Intensional Logic (TIL) system.

TIL [23], [24] is a complex system for the logical analysis of natural expressions with strong expressive power [14]. [1] Extraordinary expressiveness is another feature shared with LL [2]. Due to the absence of complete proof system, expressiveness is a barrier to full automation. Linking mentioned logic systems could solve this problem in the future.

II. The basics of TIL

TIL was founded and developed by the Czech logician Pavel Tichý in 1968 [25]. This logic allows very gentle and correct semantic-logical analysis of the natural language. In computer science, it can be used also in data modeling (HIT modeling) [7], linear programming [6] or for the development of multi-agent systems [8]. Unfortunately, TIL is relatively unknown in the world and this fact is slowing its development. Moreover, its mechanism is extremely complicated. This makes it difficult for professionals to know how to interpret and apply the disputed parts of the TIL system.

TIL is based on the concept of possible worlds. Tichý has extended Church's simple theory of types (containing a set of individuals and truth values) by modal and temporal parameters. TIL semantic scheme is:

1) the sense of expression expresses the construction,
2) the expression denotes an extension or an intension constructed by given construction,
3) extension of expression in possible world W and time point T jis the value of its intension in W, T.

Language of TIL is a modified version of the typed λ-calculus. Need to point out that syntactical matters are not principal: the TIL λ-terms serve only to depict constructions.

A. TIL as a typed logic

TIL is high-order logic with ramified theory of types [9]. Unstructured first-order type entities [13]:

1) non-functional element types:
   - o is set of the truth values (true and false),
   - ι is set of individuals (members of the universe, resp. domain) shared by all possible worlds - agents, entities...,
   - τ is set of time points alias real numbers,
   - ω is set of possible worlds,
2) functional types
   - partial functions (mappings): $(\beta\alpha_1, ..., \alpha_n)$, where $\alpha_1, ..., \alpha_n, \beta$ are types and $\alpha_1 x...x\alpha_n \rightarrow \beta$ is domain.

[1] Another important LANL system is the Montague intensional logic (MIL) [19]. We have decided not to work with MIL due to its poor versatility, bad expressiveness and lack of applicability in computer science[3], [4].

[2] This property is related to two conjunction operators and two disjunction operators of LL shown in III-A

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

The object O of type α is called α-object, we denote it O/α. Then the intensions are $((\alpha\tau)\omega)$-objects denoted as $\alpha_{\tau\omega}$-object. The extensions are not functions from possible worlds in TIL.

Examples of frequently used intensions are shown in Table I.

TABLE I
THE MOST COMMON INTENSIONS

Type	Description	Example
$o_{\tau\omega}$	Propositions	that the dog is barking
$(o\iota)_{\tau\omega}$	Properties of individuals	being smart
$\iota_{\tau\omega}$	Individual roles	the highest mountain in the world
$(\iota\iota)_{\tau\omega}$	Attributes	the president of
$(o\iota\iota)_{\tau\omega}$	Binary relations between individuals	love
$((o\iota o)_{\tau\omega})_{\tau\omega}$	Propositional attitudes	knowing that a some proposition is true
(ooo)	Binary logical operators	$\wedge\vee\rightarrow$
(oo)	Unary logical operator	$\neg$

B. TIL as a procedural logic

TIL theory uses the principle of compositionality. It says that the sense of an expression is specified by the sense of its meaningful subexpressions. Tichý identifies sense with the abstract procedures for 'calculating' sense, which he calls constructions. Constructions are structured second-order type entities. This means that constructions are made of first-order type entities and their type is $*_1$. TIL has four kinds of constructions shown in Table II, where X is any object or construction and C_i is any construction.

TABLE II
TIL CONSTRUCTIONS VS. λ-TERM.

Construction	λ-term
variable x	variable
trivialization X^0	constant
composition $[C_1...C_n]$	application
closure λxC	λ-abstraction

An informal meaning of mentioned constructions:

- Variable is primitive construction. It constructs objects dependent on their valuations.
- Trivialization is one-step procedure which takes an object X, and generates the same object.
- Composition constructs the result of an application of function to arguments.
- Closure constructs the function.

TIL uses high-order type entities where constructions are made of lower-order type entities and their type is $*_n$ (third-order type entities is $*_2$ type, etc.). Tichý makes possible to not only use constructions as constituents of other constructions, but with a ramified hierarchy of types also to insert constructions into a hyperintensional context in order to manipulate them.

Syntax of basic TIL can be described by the following grammar in BNF:

$$C ::= x \mid X^0 \mid [C_1...C_n] \mid \lambda xC \mid C_1 \wedge C_n \mid C_1 \vee C_n \mid C_1 \rightarrow C_n \mid \neg C. \tag{1}$$

C. The Parmenides principle

The way how to properly analyze natural language sentences is given by the Parmenides principle.

Let E be an expression, X an object and C a construction. If E talks about X then some subexpression of E denotes X. We can formulate Parmenides principle for TIL [15] as: An admissible analysis of E is C such that no subconstruction of C constructs X that E does not talk about.

The analysis method applied in TIL, consists of three steps:

1) type-theoretical analysis - to each object, from which E consists, is assigned a corresponding type,
2) synthesis - the final construction which represents the sense of an E is composed from individual subconstructions of E. Construction construct the object denoted by E,
3) type checking - it is being verified if the construction synthesis was correctly done (if various types are compatible and if they produce the right type of constructed object).

III. INTENSIONAL LL

J. Y. Girard introduced LL in 1987 [12] as a substructural logic in which the contraction rule and the weakening rule have their applicability restricted. LL resource-oriented characteristic allows to consider about resources as in the real world, because linear logic formulas captures problems with a limited number of sources. After using of the resources they are consumed and can not be used multiple times. It is possible trough linear logic implication $\multimap$, where an formula $M \multimap N$ is read as "consumption M provides N". For the purpose of this work, we describe syntax and semantics of the LL, the proof system we formulated in.

A. Syntax of LL

Syntax of full LL can be described by the following grammar in BNF [3]:

$$M ::= e \mid 1 \mid 0 \mid \perp \mid \top \mid M \otimes N \mid M \& N \mid M \oplus N \mid M ⅋ N \mid M \multimap N \mid M^{\perp} \mid !M \mid ?M. \tag{2}$$

where:

- metavariables M, N and elementary formula e express the literal, source, action or reaction,
- 1, 0, $\perp, \top$ are logical constants,

[3] LL can be enriched by predicates $P(t,\ldots,t)$ and by quantifiers $\forall$, $\exists$. We formulated this predicate LL in [26] and connection between it and TIL could be interesting in our future research.

- connections $\otimes, \parr$ are intensional conjunction and intensional disjunction,
- connections $\oplus, \&$ are extensional conjunction and extensional disjunction,
- $\multimap$ is linear implication,
- $(.)^{\perp}$ is linear negation,
- exponential $!M$ ("of course") as the exponential conjunction of M and $?M$ ("why not") as the exponential disjunction of M.

In the following subsection III-B is shown that for the purposes of this article, it is not necessary to work with the full LL but only with its fragment.

B. Semantics of LL

The overview of possible LL fragments [16] is shown in Figure 1.

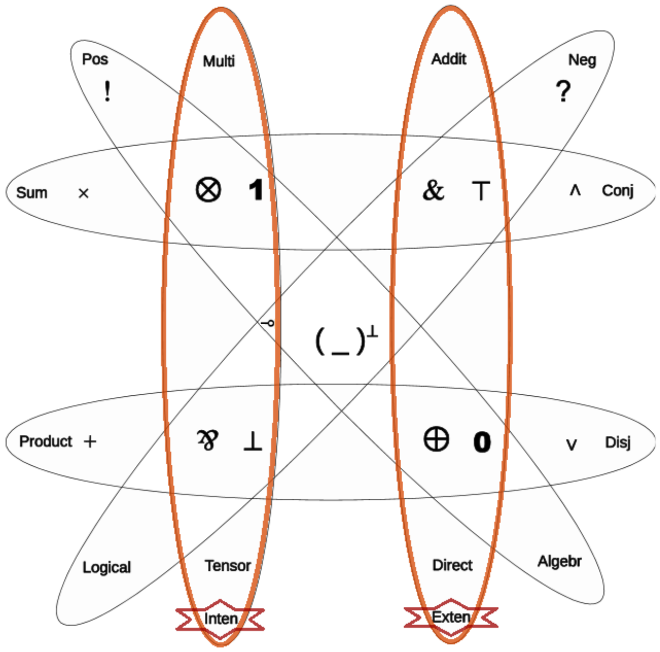


Fig. 1. Fragmentation of LL

In favour of this article, it is sufficient to focus on next two fragments of LL:

- extensional fragment - Tarski's semantic tradition,
- intensional fragment - Heyting's semantic tradition.

Tarski's approach deals with the meanings (reference) of formulas. Truthfulness of action can be determined by evaluating the truth values of formulas. Logical constant $\top$ is used for linear true **0** for linear false. Extensional fragment of LL can be described with the following syntax:

$$M \ = e \mid 0 \mid \top \mid M \ \& \ N \mid M \oplus N \mid M \multimap N \mid M^{\perp}. \qquad (3)$$

On the other hand, Heyting's approach deals with the sense of formulas. The sense represents the way in which is the expression connected to its meaning. Intensional fragment of LL can be described as follows:

$$M \ = e \mid 1 \mid \perp \mid M \parr N \mid M \otimes N \mid M \multimap N \mid M^{\perp}. \qquad (4)$$

Due to the fact that TIL has intensional character, we use intensional fragment of LL. Therefore we do not consider about truthfulness of the sentence in the natural language, but its validity. An informal meaning of intensional linear connectives:

- logical constant **1** is neutral element for $\otimes$ and represents a linear sense. Constant $\perp$ is neutral element for $\parr$ and represents a linear nonsense,
- binary intensional linear disjunction $M \parr N$ expresses that if an action M is not performed, then an action N is done or vice versa,
- binary intensional linear conjunction $M \otimes N$ expresses that both actions M and N will be performed simultaneously,
- binary causal linear implication $M \multimap N$ expresses that resource is M consumed and it becomes a linear negation $M^{\perp}$.

IV. Applying LL principles to a TIL constructions

The TIL and LL logic systems can be linked through their intensional character. After translation of the intensional LL fragment to TIL, its syntax will be described by the following BNF grammar:

$$C ::= x \mid X^0 \mid [C_1 ... C_n] \mid \lambda x C \mid C_1 \otimes C_n \mid C_1 \parr C_n \mid C_1 \multimap C_n \mid C^{\perp}. \qquad (5)$$

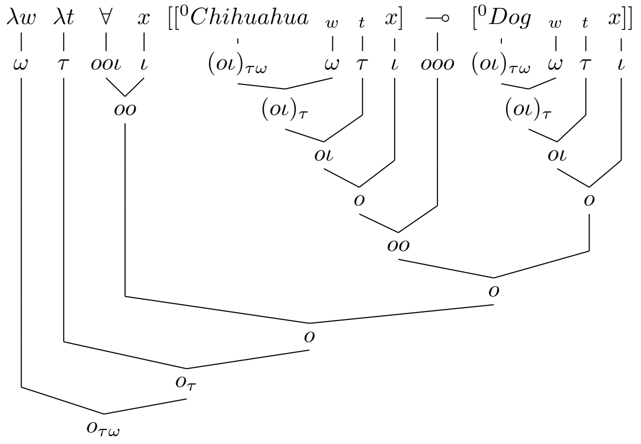
We can demonstrate the use of TIL enriched with LL principles in the following examples:

- All chihuahuas are dogs. - using of $\multimap$ operator,
- PLuto is a small planet or a large dwarf. - using of $\parr$ operator,
- Ema likes friend of Alex, but she does not love him. - using of $\otimes,^{\perp}$ operators.

We illustrate the method of analysis by analyzing chosen sentences. Sentences are analyzed in three steps - type analysis, synthesis and type checking. A sentence, the sense of which is correctly expressed, is always a type $o_{\tau\omega}$.

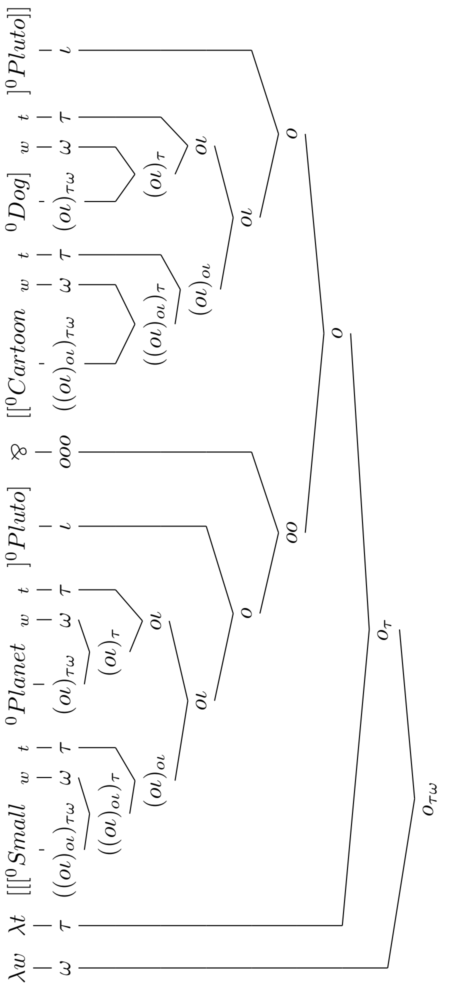
A. All chihuahuas are dogs.

1) Type analysis:
   All/$oo\iota$, Chihuahua/$(o\iota)_{\tau\omega}$, Dog/$(o\iota)_{\tau\omega}$, x/ι, $\multimap$/ooo
2) Synthesis:
   λwλt $\forall$x [[^{0}Chihuahua$_{wt}$ x] $\multimap$ [^{0} Dog$_{wt}$ x]]
3) Type checking:


λw λt ∀ x [[⁰Chihuahua w t x] ⊸ [⁰Dog w t x]]
ω τ οοι ι (οι)τω ω τ ι οοο (οι)τω ω τ ι
οο (οι)τ (οι)τ
οι οι
ο ο
οο
ο
ο
οτ
οτω

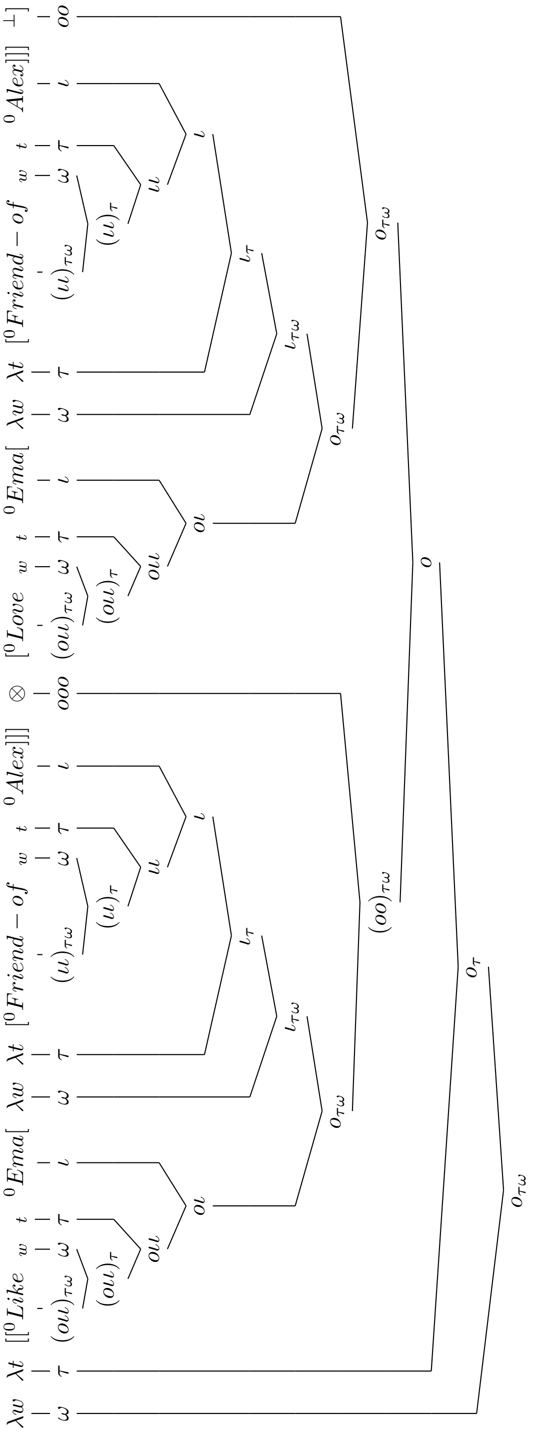
B. Pluto is a small planet or a cartoon dog.

1) Type analysis:
   Pluto/ι, Small/$((o\iota)o\iota)_{\tau\omega}$, Planet/$(o\iota)_{\tau\omega}$, Cartoon/$((o\iota)o\iota)_{\tau\omega}$, Dog/$(o\iota)_{\tau\omega}$, ⅋/$ooo$
2) Synthesis:
   λwλt [[[^{0}Small$_{wt}$ ^{0}Planet$_{wt}$]^{0}Pluto] ⅋ [[^{0}Cartoon$_{wt}$ ^{0}Dog$_{wt}$]^{0}Pluto]]
3) Type checking:


λw λt [[[⁰Small w t ⁰Planet w t]⁰Pluto] ⅋ [[⁰Cartoon w t ⁰Dog w t]⁰Pluto]]
ω τ ((οι)οι)τω ω τ (οι)τω ω τ ι οοο ((οι)οι)τω ω τ (οι)τω ω τ ι
((οι)οι)τ (οι)οι (οι)τ οι οι ο οο
ο
οτ
οτω

C. Ema likes friend of Alex, but she does not love him.

1) Type analysis:
   Ema/ι, Like/$(o\iota\iota)_{\tau\omega}$, Friend_of/$(\iota\iota)_{\tau\omega}$, Alex/$\iota$, Love/$(o\iota\iota)_{\tau\omega}$, ⊗/$ooo$, $^{\perp}$/oo
2) Synthesis:
   λwλt [[^{0}Like$_{wt}$ ^{0}Ema [^{0}Friend_of$_{wt}$ ^{0}Alex] ⊗ [^{0}Love$_{wt}$ ^{0}Ema [^{0}Friend_of$_{wt}$ ^{0}Alex]]$^{\perp}$]]
3) Type checking:


λw λt [[⁰Like w t ⁰Ema[λw λt [⁰Friend−of w t ⁰Alex]]] ⊗ [⁰Love w t ⁰Ema[λw λt [⁰Friend−of w t ⁰Alex]]] ⊥]
ω τ (οιι)τω ω τ ι ω τ (ιι)τω ω τ ι οοο (οιι)τω ω τ ι ω τ (ιι)τω ω τ ι οο
(οιι)τ οιι οι (ιι)τ ιι ι ιτ ιτω οτω (οο)τω οτω ο
οτ
οτω

V. Conclusion

In this article, we focused on two LANL approaches - Tichý's TIL and Girard's LL. After their brief analysis, it is clear that the chosen logic systems are very different. The fact that both represent highly effective LANL tools stimulates us to find connections between them. The primary link is the intensional character of both the investigated mechanisms. We have replaced TIL logical operators with source-oriented LL operators. However, the properties of the original TIL have been preserved, as demonstrated by the examples of the analyzed sentences. LL operators have expanded the TIL's expressiveness because they capture consumption of resources.

In future research, it is interesting to focus on the opposite process - the integration of TIL principles into LL. This process would involve the transformation of constructions into linear formulas and the exact formulation of syntax, semantics, and LL proof calculus, enriched with TIL principles. This extended LL mechanism could be explored both theoretically and practically. From the theoretical point of view, it is interesting to explore natural language dialogues using the Ludics theory. An innovative, practical application would be to extend the source-oriented interpreter Vorvan [2] through constructions.

Acknowledgment

This work was supported by KEGA Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under Grant No. 077TUKE-4/2015 "Promoting the interconnection of Computer and Software Engineering using the KPIkit". This support is very gratefully acknowledged.

References

[1] A. Avron: *The Semantics and Proof Theory of Linear Logic* In: Theoretical Computer Science, Vol. 57, 1988, pp. 161-184

[2] K. Bilan, D. Mihályi, *Logic Programming Paradigm Language Interpreter Based on Resource-oriented Logical System Principles*, In: Electrical Engineering and Informatics 7 : Proceedings of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, 2016, pp. 291-296, ISBN 978-80-553-2599-6.

[3] Z. Bilanová, M. Uchnár, *Comparison of the approaches of Montague and Tichý within a logical analysis of an English sentence*, In: POSTER 2017. - Prague : Czech Technical University, 2017, pp. 1-6, ISBN 978-80-01-06153-4.

[4] Z. Bilanová, *Applications of transparent intensional logic and montague intensional logic on natural language sentences: A Review*, In: SCYR 2017. - Košice : TU, 2017, pp. 32-35, ISBN 978-80-553-3162-1.

[5] T. Brauner, *Introduction to Linear Logic* BRICS LS-96-6, 1996, pp. 52, ISSN 1395-2048

[6] M. Číhalová, N. Ciprich, M. Duží, M. Menšík, *TIL and Logic Programming* In: Raslan 2008, eds. P. Sojka, Masarykova univerzita, Brno, Vol. 33, 2008, pp. 17-30, ISSN 9788021047419.

[7] M. Duží, *Logical Foundations of Conceptual Modelling Using HIT Data Model* In:Information Modelling and Knowledge Bases XII, IOS Press, 2001, pp. 65-80, ISBN 1586031635.

[8] M. Duží *TIL as the Logic of Communication in a Multi-Agent System* In: Research in Computing Science, Vol. 33, 2008, pp. 27-40, ISSN 1870-4069.

[9] M. Duží *Deduction in TIL: From simple to ramified hierarchy of types* In: Organon F, Vol. 20, No. 2, pp. 5-36, 2013.

[10] M.-R. Fleury, M. Quatrini, S. Troncon, *Dialogues in Ludics* Logic and Grammar: Essays Dedicated to Alain Lecomte on the Occasion of His 60th Birthday, 2011, pp. 159, ISBN 9783642214899.

[11] Ch. Fouquere, M. Quatrini, *Ludics and Natural Language: First Approaches* In: International Conference on Logical Aspects of Computational Linguistics LACL 2012: Logical Aspects of Computational Linguistics, 2012, pp. 21-44, ISSN 1611-3349.

[12] J.-Y. Girard, *Ludics and Natural Language: First Approaches* In: International Conference on Logical Aspects of Computational Linguistics LACL 2012: Logical Aspects of Computational Linguistics, 2012, pp. 21-44, ISSN 1611-3349.

[13] A. Horák, *Types in Transparent Intensional Logic and Easel - a Comparison* In: Proceedings of the IASTED International Conference Artificial Intelligence and Applications 2004. Anaheim, Calgary, Zurich: The International Association of Science and Technology for Development, 2004, pp. 833-837, ISBN 0-88986-404-7.

[14] P. Materna, *Expresivita logické analýzy přirozeného jazyka. [Expressivity of logical analysis of natural language.]* In: Organon F, Vol. 20, No. 2, 2013, pp. 112-116.

[15] P. Materna, M. Duží, *Parmenides principle (The Analysis of Aboutness)* In: Philosophia, Vol. 32, No. 1-4, 2005, pp. 155-180.

[16] D. Mihályi, V. Novitzká, *What about linear logic in computer science?* In: Acta Polytechnica Hungarica, Vol. 33, No. 1, 2014, pp. 61-78, ISSN 1785-8860.

[17] D. Mihályi, V. Novitzká, *Towards to the Knowledge in Coalgebraic model IDS* In: Computing and Informatics, Vol. 10, No. 4, 2013, pp. 147-160, ISSN 1335-9150.

[18] D. Mihályi, V. Novitzká, V. Slodičák, *Towards to the Knowledge in Coalgebraic model IDS* In: Proceedings of CSE 2008 International Scientific Conference on Computer Science and Engineering. - Košice : TU FEI, 2008, pp. 48-56, ISBN 9788080860929.

[19] R. Montague, *The Proper Treatment of Quantification in Ordinary English* K. J. J. Hintikka, J. M. E. Moravcsik, and P. Suppes (eds.), Approaches to Natural Language (Synthese Library 49), vol. Dordrecht: Reidel, 1973, pp. 221–242.

[20] A. Lecomte, M. Quatrini *Ludics, dialogue and inferentialism*, In: The Baltic International Yearbook of Cognition, Logic and Communication, Vol. 8, 2013, pp. 1-33, ISSN 1944-3676.

[21] A. Lecomte, *Ludics and Its Applications to Natural Language Semantics*, In: International Workshop on Logic, Language, Information, and Computation, WoLLIC 2009: Logic, Language, Information and Computation, 2009, pp. 242-255, ISSN 1611-3349.

[22] J. Perháč, V. Novitzká, D. Mihályi, *Between syntax and semantics of resource oriented logic for IDS behavior description* In: Journal of Applied Mathematics and Computational Mechanics, Vol. 15, No. 2, 2016, pp. 105-118, ISSN 2353-0588.

[23] J. Raclavský, *Jména a deskripce: logicko-sémantická zkoumání* Czech Republic: Olomouc, pp. 396, 2009, ISBN 978-80-7182-277-6.

[24] P. Tichý, *The Foundations of Frege's Logic* 1nd ed, Berlin and New York: De Gruyter, pp. 303, 1988, ISBN 9783110849264.

[25] P. Tichý, *Smysl a procedur* In: Filosofický časopis, Vol. 16, 1968, pp. 222-232.

[26] L. Vokorokos, Z. Bilanová, D. Mihályi, *Hanoi towers in resource oriented perspective*, In: SAMI 2017. - Danvers : IEEE, 2017, pp. 179-184, ISBN 978-1-5090-5654-5.

Low-level computer vision techniques for processing of extensive air shower track images

Michal Vrábel, Ján Genči
Faculty of Electrical Engineering and
Information technologies,
Technical University of Košice,
Letná 9, 04001 Košice, Slovakia
Email: michal.vrabel@tuke.sk, jan.genci@tuke.sk

Pavol Bobík
Institute of Experimental Physics
Slovak Academy of Sciences,
Watsonova 47, 04001 Košice, Slovakia
Email: bobik@saske.sk

***Abstract*—An extensive air shower can be observed as a bright spot moving through the field of view of an orbital fluorescence detector. A challenging part of the air shower recognition is segmentation of its track. The issues arise from a low signal to noise ratio. This paper provides a short review of selected low-level computer vision techniques such as filtering and thresholding methods, which are for a demonstration applied to a composite simulated air shower image. The article should provide a shortlist of algorithms that can be applied as a part of more complex event classification or reconstruction procedure.**

I. Introduction

The task of the fluorescence air shower detectors related to a planned detector JEM-EUSO (Japanese Experiment Module - Extreme Universe Space Observatory) [1] is to observe extensive air shower track from top-down perspective. The air shower is produced when ultra high energy cosmic ray particle collides with particles of the atmosphere. The shower track is expected to be mostly observable in near UV spectrum as a bright spot moving through field of view of a detector on a straight track in approximately speed of light. An "EUSO-class" detector is a telescope consisting of Fresnel-lens optics attached to a Multi anode photomultiplier (MAPMT) matrix grouped in photo-detector module (PDM). Such detector consists of one or more PDM modules. The module has 36 MAPMT grouped in sets of four into an elementary cell (EC). More in-depth description of the instruments can be found in [2]. The first operational EUSO detectors [3]–[5] consist of a single PDM with a resolution of 48 × 48 pixels. Although, it is an atypical imaging instrument, data can be generally considered as a sequence of bitmaps. However the simplification is disregarding real pixel positions on focal surface, which induces noticeable deformations in case of full 137 PDM detector design.

To find air shower events, the pattern is searched in experiment data. Then correctly classified events are analyzed in with an event reconstruction procedure to estimate primary particle arrival direction and energy. Typically a shower track can be analyzed as a sequence of frames, or in computationally less demanding way, using composite projections of shower in X-Y, Time-X, Time-Y planes. The composite is constructed by selecting a maximum value of the pixel within the time window. The approach succeeds when air shower track intensity is higher than background intensity fluctuations within the time window, otherwise even shower track part recognizable in a single frame.

Sonka et al. [6] organized the computer vision techniques into *low-level image processing* (*low-level vision*) and *high-level image understanding*. Davies [7] adds an additional layer - *intermediate-level vision*, which is concerned with a feature extraction of basic features, such are line orientations, circle centers, object boundaries, or basic pattens. Higher level vision then infers an object class based on image features.

The following sections will provide overview of some low-level image processing algorithms, mainly focusing on those, that are deemed applicable to data captured by an "EUSO-class" detector. Selected methods will be demonstrated on examples of composite X-Y projections of a shower.

II. Low-Level Computer Vision Methods

The low-level vision approaches handle the first step of the image pattern recognition. Most of the techniques that fit in this category are better characterized as a digital image processing. The methods can provide basic normalization of the data and initial feature extraction. The latter of the applications is the main motivation to utilize these methods in processing of EUSO events.

Davies [7] identifies following low-level technique categories:

- **Image processing and basic filtering** - The operations range from adjustment of brightness to smoothing and noise reduction. Selected examples of these approaches will be reviewed in the section III.
- **Thresholding** - The methods realize segmentation of an image often into two classes - foreground and background separated by a threshold value. Additionally, this is a data reduction technique, which can limit color depth of an image using one or more threshold values. Selected examples of the thresholding methods will be reviewed in the section IV.
- **Edge detection** - The methods realize another form of segmentation that is aimed at detecting pixels belonging to an edge and locating steep intensity gradients in an

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

image. Similarly as thresholding, edge detection provides data reduction. Edge detectors often use convolutional masks approximating first or second derivation. The first derivative operators can be used as template matching [8]–[10] or differential gradient operators [8], [11], [12], but both indicate edges by a function gradient at a given point. On the other hand, the second derivative operations [13] indicate edges by ”zero crossings”. More sophisticated methods for edge detection can use active contours method [14] or graph cut method [15].

- **Corner and interest point detection** - Similar to the previous, but these methods are designed to give a response in locations of an image where intensity gradients create an edge [16], [17], or methods exploit properties of median filters [18].
- **Basic texture analysis** - The methods exploit basic principles of filtering to extract quantifiable information about textures.

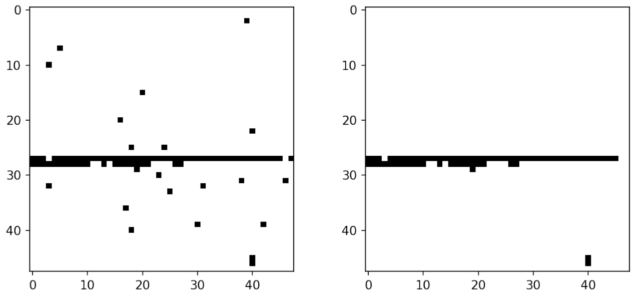
III. Image Processing Operations

Image processing operations are usually applied to a single channel image. The most simple of them are increasing brightness, or image contrast. To discriminate background noise in the detector data, the simplest filtering operation is to expect maximum noise level to be under certain predefined threshold, for instance the mean value of a noise distribution. Same principle can be applied to images in general, creating binary images.

A. Binary images

One of the simplest operations on images is to threshold the pixel intensity and consider data just as *binary images* [7], [19]. Each pixel with intensity under the threshold is 0 (white), and 1 (dark) otherwise. Dark objects in such images can be shrinked, expanded, or transformed into contours, by very simple procedures which look at immediate neighborhood of a pixel and count number of dark or light pixels. Another simple technique is ”salt and pepper” noise reduction, setting pixel to 0 if none of the pixels in the neighborhood has value 1, or setting it to 1 in the opposite case. These operations can be generalized as morphological operations of dilation and erosion, which alongside input image require kernel, that defines mask used for expansion or shrinking.

Especially ”pepper” removal procedure can be applied in processing EUSO detector data to additionally remove fluctuations not filtered-out by a thresholding procedure. The ”pepper” removal algorithm, described in book by Davis [7], sums a central pixel’s neighboring pixel values. If the sum is equal to zero, then the value of the central pixel in an output image is set to zero, otherwise the source image value is used. An application of the algorithm is shown on figure 1. An issue with such simple implementation arises when thresholding does not select continuous blocks of pixels, then more complex neighborhood selection procedure is required.



(a) Original

(b) After ”pepper” removal

Fig. 1. Composite shower track as a binary image.

B. Noise filtering

Operations in this category are filters intended for noise removal. In image processing, they are usually applied as a step of more complex processing. Applicability the methods for shower recognition is limited because typically even composite images of a shower don’t have sufficiently high signal-to-noise ratio. Also, sometimes it is important to consider, that applying such filters induces shifts in an output image.

Figure 2 presents a selection of filter functions applied to a composite image of a simulated air shower with $\approx 35^\circ$ inclination and average values of signal pixels are $\approx 13\,\%$ above average background considering top $50\,\%$ of pixel intensities. If not specified, the presented filters use implementation of the scikit-image library [20].

1) Gaussian smoothing: This method, also referred as Gaussian blur [21], is implementation of a low-pass filter in the spatial domain, exploiting convolutional theorem [22], stating that a convolution of a signal with a Fourier transform of a function in the spatial domain is equal to multiplication of the signal by the function in the spatial frequency domain. A Fourier transform of a Gaussian function is another Gaussian function Practically the filter is applied as a convolution of Gaussian function approximation with an input image.

Possible application of the method is to smooth background (see figure 2b), create weights distributions, and sharp-unsharp masking used to filter-out large proportion of background noise (see figure 2c).

2) Median filters: Purpose of median filters [21] is to replace image pixels that have extreme intensities. The expectation is that such peaks are noise. The method is applied by examining local neighbourhood of each pixel and removing points with extreme intensities. The operation of the filter requires construction of pixels local neighbourhood intensity histogram, that is used for determination of median. The filter tends to remove a shower track from an image (see figure 2d) therefore it is not directly ideal for segmentation, but on the other hand this might be useful for creation of background model.

3) Rank order filters: These filters take and increasingly order all intensity values (n) in the local neighborhood, then

r-th ($r \leq n$) value is selected as the filter's output [7] (see figures 2e and 2f).

4) Wiener filter: An example of more sophisticated signal processing filter is the Wiener filter [23] or more precisely its application - Wiener deconvolution. The filter considers measurements which contains response of measurement equipment and noise. The Wiener deconvolution is a function that minimizes mean square error between estimated and desired signal. It applies deconvolution of a system impulse response (point spread function) that is attenuated with increasing noise. The presented example on figure 2g uses simplified implementation provided by SciPy library [24] which assumes constant power spectrum of the signal and white noise, that is estimated using average of local variance and input.

5) Wavelet transform filter: The filter utilizes property of noise in the wavelet domain of an image. Wavelet thresholding [25] is applied to remove random noise represented by small values in the wavelet domain.

C. Gabor filter

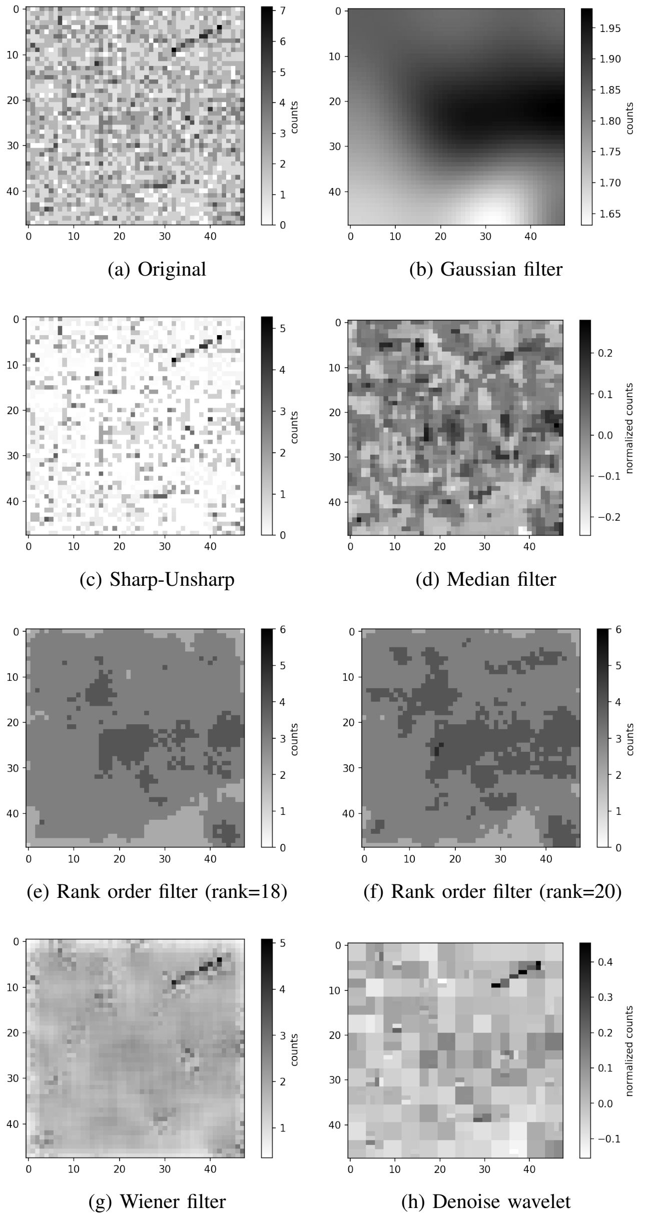
The Gabor filter [26] consist of convolving a Gabor function (also referred as a gabor wavelet) with an image in the spatial domain, therefore based on the convolution theorem, the function allows to identify specific frequency content in an image. Illustration of the filter is on figure 3. The approach can be also categorized as a texture analysis method because in practice different Gabor kernels are applied to an image, which provides extraction of features describing texture presence in an image. This principle can be also related to visual cortex operation.

The method has been tested to provide classification of the shower tracks and ”empty” background frames. Considering only single air shower X-Y projection image and single background frame image from EUSO-SPB flight, classification is done by calculating sum of squared errors of mean and variance between the ”training” and evaluated ”test” image. If ”training” is done using image on figure 2a and dataset described in the section IV, even this very simple classification technique is able to provide approximately 65% precision. The precision value is mainly impacted by high true negative count.

IV. Thresholding

The simplest clue towards event detection, by any kind of a pixel detector, is raise in pixel intensity (in case of photomultipliers, increase of the photoelectron counts value). Thresholding can serve as a basic form of segmentation, dividing the pixel intensities by a preselected threshold value, therefore classifying the image pixels into background and foreground.

Determination of thresholds for background rejection is of a great importance for shower pattern recognition methods. The simplest method of selecting the shower track is to simply filter out everything under a certain threshold value. This basic technique might be the first step of more complex track finding method. Although, varying background intensity distribution in an image, between frames of a sequence, or between event



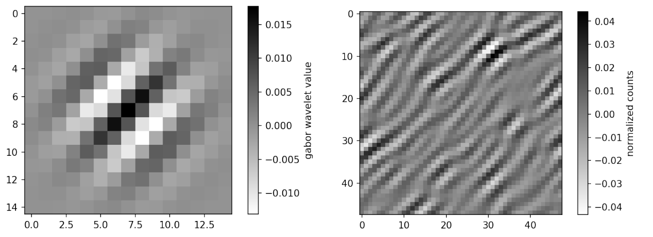
(a) Original (b) Gaussian filter

(c) Sharp-Unsharp (d) Median filter

(e) Rank order filter (rank=18) (f) Rank order filter (rank=20)

(g) Wiener filter (h) Denoise wavelet

Fig. 2. Examples of filters applied to a composite shower image.



(a) Gabor wavelet freq $= 0.25$, $\theta = 45°$, $\sigma = 3$

(b) Gabor wavelet applied to the image

Fig. 3. An example of Gabor filter applied to a composite shower image.

samples in a dataset requires less configuration dependent method of a threshold determination.

Challenge in application of thresholding algorithms from field of computer vision to air shower track recognition is small difference between local peak values of background and intensities of a shower track.

The threshold determination is problem tackled since beginnings of the computer vision field. Very comprehensive survey has been written by Sezgin and Sankur [27]. The review's authors categorize the techniques into:

- **Histogram shape-based methods** - Histogram of the intensity distribution is analyzed, to find peaks and valleys.
- **Clustering-based methods** - The thresholding is viewed as a clustering problem dividing dataset into two (or more) classes.
- **Entropy-based methods** - The aim is to obtain greatest order of a system by applying a threshold separating intensity probability distributions.
- **Object attribute-based methods** - A threshold is selected by maximizing a measure of similarity between the original gray level and created binary images.
- **Spatial methods** - The methods use the higher-order probability distribution and correlation between pixels. Not only the pixel intensity distribution is utilized but also dependency of pixels in a neighborhood.
- **Locally adaptive methods** - The methods adapt threshold to local intensity distribution, either by modeling background, splitting image to independent subimages, determining local threshold of each pixel, or combination of the last two, using overlapping subimages [28]. Correct functionality of the method is dependent on balance between a subimage dimensions, a sample size, and an overlap size.

Figure 4 demonstrates examples of some thresholding methods. Numbers in brackets denote percentage of selected simulated and background pixels count. Presented thresholding methods use scikit-image library's implementation [20] if available, otherwise implementation inspired by the ImageJ Auto Threshold plug-in is used [29], [30]. Actually, several of the algorithms implemented in scikit-image are based on ImageJ's implementation. Methods in the figures are classified in the following subsections of this section.

Table I compares several thresholding methods applied to dataset of ≈ 4000 simulated EUSO-SPB events with actual UV background observed during the flight. All of the analyzed composite event tracks have uniformly distributed higher inclination than $15°$, at least 3 pixels triggered using JEM-EUSO's L1 triggering algorithm [31], and at least on 5 frames of the event true signal intensity was higher than background. The background intensity distribution in each X-Y projection was ”flattened” by subtracting X-Y projection of average pixel values within 32 frame window of background observation and only positive values were considered. The distribution of background intensities can be described by an exponential function $29\,056.54 \exp(-1.66x) - 27.23$, if values are in 5 bins.

TABLE I
THRESHOLDING APPLIED TO X-Y AIR SHOWER PROJECTIONS.

	$\overline{t_{bg}/a_{bg}} \pm \sigma(t_{bg}/a_{bg})$	$\overline{t_{sig}/a_{sig}} \pm \sigma(t_{sig}/a_{sig})$
Mean	0.403 ± 0.035	0.690 ± 0.078
Mean $\times 2$	0.016 ± 0.008	0.313 ± 0.100
Minimum	0.007 ± 0.057	0.121 ± 0.109
Intermodes	0.004 ± 0.026	0.201 ± 0.102
Yen	0.026 ± 0.029	0.333 ± 0.101
Triangle	0.223 ± 0.135	0.533 ± 0.163
Isodata	0.109 ± 0.122	0.436 ± 0.151
Otsu	0.183 ± 0.140	0.487 ± 0.174

The first column of values in the table describes arithmetic mean value of selected background pixels count (t_{bg}) and all background pixels count (a_{bg}) ratio, the aim of a thresholding method is to make this value as low as possible. The second column of values describes arithmetic mean value of selected true signal pixels count (t_{sig}) and all true signal pixels count (a_{sig}) ratio, the aim of a thresholding method is to make this value as high as possible. Second values in the columns are standard deviations.

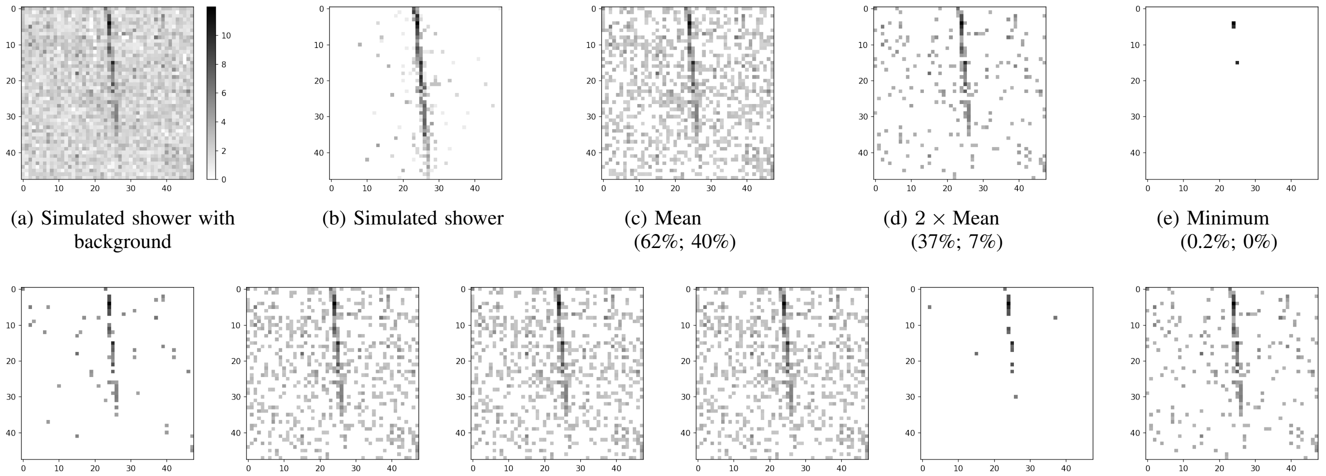
One of the simplest methods for thresholding consist of just calculating mean value of pixel intensities (see figure 4c). Additional multiplication of the mean with a constant (see figure 4d) can be helpful if ratio of feature to background pixels count is low. Due the method simplicity, we are not classifying it into specific category.

A. *Histogram shape-based methods*

In simple situations, a threshold can be determined by analysis of the intensity levels histogram and search of significant minimum, which is interpreted as a threshold value [32]. It is assumed that there are visible peaks in intensity distribution between foreground and background. Location of minimum is subject of major difficulties, caused by width of a valley, multiple minima, noise, or inherent multimodality of the histogram. Several such approaches has been reviewed by Glasbey [33].

Minimum method of thresholding [34] (see figure 4e) attempts to smooth histogram using running average of size 3. The smoothing procedure ends when there are only two maxima and then global minimum is selected as a threshold. Method *intermodes* uses an average value between selected peaks. Weakness of these methods is requirement of eventual histogram's bimodality, which might not be satisfied, also manual page for ImageJ's implementation [30] mentions problems with broad flat valleys and extremely unequal peaks.

Triangle thresholding method [35] (see figure 4g) has been designed assuming an image without objects (background) subtracted from an analyzed image, it's goal was only to ”account for variations in staining intensity”. The method assumes background pixels to be much more prevalent in an image and an intensity histogram is not expected to be bimodal. Actually, the shape after normalization should serve as legs of a right angled triangle. The longest line segment



(a) Simulated shower with background

(b) Simulated shower

(c) Mean (62%; 40%)

(d) 2 × Mean (37%; 7%)

(e) Minimum (0.2%; 0%)

(f) Yen (17%; 2%)

(g) Triangle (49%; 25%)

(h) Isodata (52%; 28%)

(i) Otsu (53%; 28%)

(j) Intermodes (8%; 0.1%)

(k) Max entropy (35%; 7%)

Fig. 4. Examples of filters applied to composite shower image.

perpendicular to the hypotenuse and connected to histogram is selected as a base for the threshold.

B. Clustering-based methods

When a peak representing a feature is small, or even hidden between noise, the determination of a threshold by a local minimum becomes impracticable. Methods in this category cluster pixels into foreground and background classes. For example method by Ridler [36] divides pixels into two classes and integrates their pixel intensities separately, then means of the classes are averaged to determine a threshold. Procedure continues until the threshold stops changing. Thresholding method *Isodata* (see figure 4h) is based on Ridler's algorithm, but starts from threshold value 0. In the variance-based thresholding, the aim is to determine threshold which optimizes criteria based of the intra-class (within-class), inter-class (between-class), and total variance. Popular approach is the *Otsu* thresholding [37] (see figure 4i). If a single threshold is required, the task of the algorithm is to maximize inter-class variance, or minimize of the intra-class variance, in other words creating clusters of similar values.

C. Entropy-based methods

Another way of a threshold selection is to consider entropy of data. It is high if the variable values are well distributed over an available range, and low if they are narrowly distributed and well ordered.

In the *Max entropy* method [38] (see figure 4k), the aim is to select a threshold that maximizes a sum of foreground and background entropies. *Yen* thresholding method [39] (see figure 4f) uses entropic correlation, aimed at decreasing the number of operations for determining a threshold value.

CONCLUSION

In this paper we have reviewed selected low-level computer vision techniques and applied them to composite X-Y projections of simulated air showers captured on a single photo detector module. The X-Y projections were constructed using background with varying intensity.

The example of the ”pepper removal” procedure applied on a thresholded binary image of a composite shower track motivates that ”pepper removal” procedure (or similar) could be used in conjunction with less restrictive thresholding method, but preserving most of a track. Out of the tested filtering methods (also those not included in this paper), there is possible applicably for several. Wiener filter method stands out, and in future we would like to investigate the method in depth with better models of noise and point spread function.

Multiple thresholding methods were tested - depending on a requirement different thresholding might be suitable in the future. Examples and quantitative analysis show that these established computer vision methods are practicably applicable for air shower recognition. A challenge for thresholding might be narrow distribution of values, thus sometimes even rounding plays an important role.

Several methods in this review can be implemented using convolutional kernels, including the Gabor filter method. This motivates that classification or reconstruction method by convolutional neural network [40]–[42] might be successful because this type of networks actually constructs such kernels as a result of the learning.

Topics opened in this paper should be continued with testing of more noise reduction techniques, more thresholding methods, and edge detection algorithms.

Acknowledgment

This work was supported by project KEGA 047TUKE-4/2016 Integrating software processes into the teaching of programming.

References

[1] A. Haungs, G. Medina-Tanco, and A. Santangelo, "Special issue on the JEM-EUSO mission," *Experimental Astronomy*, vol. 40, no. 1, pp. 1–2, 2015.

[2] JEM-EUSO Collaboration, J. H. Adams, S. Ahmad, J. N. Albert, D. Allard, L. Anchordoqui, V. Andreev, A. Anzalone, Y. Arai, K. Asano, M. Ave Pernas, P. Baragatti, P. Barrillon, T. Batsch, and J. Bayer, "The JEM-EUSO instrument," *Experimental Astronomy*, vol. 40, no. 1, pp. 19–44, 11 2015.

[3] J.-E. Collaboration, J. H. Adams, S. Ahmad, J. N. Albert, D. Allard, and L. Anchordoqui, "The EUSO-Balloon pathfinder," *Experimental Astronomy*, vol. 40, no. 1, pp. 281–299, 2015.

[4] L. Wiencke, "EUSO-Balloon mission to record extensive air showers from near space," in *Proceedings, 34th International Cosmic Ray Conference (ICRC 2015): The Hague, The Netherlands, July 30-August 6, 2015*, vol. ICRC2015, 2016, p. 631. [Online]. Available: http://pos.sissa.it/archive/conferences/236/631/ICRC2015_631.pdf

[5] M. Ricci, M. Casolino, and P. Klimov, "Mini-EUSO: a pathfinder for JEM-EUSO to measure Earths UV background from the ISS." in *Proceedings, 34th International Cosmic Ray Conference (ICRC 2015): The Hague, The Netherlands, July 30-August 6, 2015*, vol. ICRC2015, 2016, p. 599. [Online]. Available: http://pos.sissa.it/archive/conferences/236/599/ICRC2015_599.pdf

[6] M. Sonka, V. Hlavac, and R. Boyle, *Introduction*. Boston, MA: Springer US, 1993, pp. 1–12.

[7] E. Davies, *Computer and Machine Vision: Theory, Algorithms, Practicalities*, ser. Academic Press. Elsevier, 2012.

[8] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.

[9] R. A. Kirsch, "Computer determination of the constituent structure of biological images," *Computers and Biomedical Research*, vol. 4, no. 3, pp. 315 – 328, 1971.

[10] G. S. Robinson, "Edge detection by compass gradient masks," *Computer Graphics and Image Processing*, vol. 6, no. 5, pp. 492–501, oct 1977.

[11] L. G. Roberts, "Machine perception of three-dimensional soups," Ph.D. dissertation, Massachusetts Institute of Technology, 1963.

[12] R. O. Duda, P. E. Hart, D. G. Stork *et al.*, *Pattern classification*. Wiley New York, 1973, vol. 2.

[13] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980. [Online]. Available: http://rspb.royalsocietypublishing.org/content/207/1167/187

[14] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan 1988. [Online]. Available: https://doi.org/10.1007/BF00133570

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[16] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[17] P. R. Beaudet, "Rotationally invariant image operators," pp. 579–583, 1979, cited By 64.

[18] "Local ordered grey levels as an aid to corner detection," *Pattern Recognition*, vol. 17, no. 5, pp. 535 – 543, 1984.

[19] M. Sonka, V. Hlavac, and R. Boyle, *Data structures for image analysis*. Boston, MA: Springer US, 1993, pp. 42–55.

[20] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014.

[21] M. Nixon and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision, Third Edition*, 3rd ed. Academic Press, 2012.

[22] E. W. Weisstein, "Convolution theorem," from MathWorld–A Wolfram Web Resource, Accessed: 2017-09-20. [Online]. Available: http://mathworld.wolfram.com/ConvolutionTheorem.html

[23] N. Wiener, "Extrapolation, inerpolation, and smoothing of stationary time series," 1975.

[24] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed 2017-09-20]. [Online]. Available: http://www.scipy.org/

[25] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, Sep 2000.

[26] J. R. Movellan, "Tutorial on Gabor filters," 2005, technical report MPLab Tutorials UCSD MPLab, Accessed: 2017-09-20.

[27] M. Sezgin *et al.*, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic imaging*, vol. 13, no. 1, pp. 146–168, 2004. [Online]. Available: http://pequan.lip6.fr/~bereziat/pima/2012/seuillage/sezgin04.pdf

[28] C. Chow and T. Kaneko, "Automatic boundary detection of the left ventricle from cineangiograms," *Computers and Biomedical Research*, vol. 5, no. 4, pp. 388–410, aug 1972.

[29] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri, "The imagej ecosystem: An open platform for biomedical image analysis," *Molecular reproduction and development*, vol. 82, no. 7-8, pp. 518–529, 2015.

[30] G. Landini, "Auto threshold," 2009, accessed: 2017-09-20. [Online]. Available: http://imagej.net/Auto_Threshold

[31] G. Abdellaoui, S. Abe, A. Acheli, J. H. Adams *et al.*, "Cosmic ray oriented performance studies for the JEM-EUSO first level trigger," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 866, pp. 150–163, 2017.

[32] J. S. Weszka, "A survey of threshold selection techniques," *Computer Graphics and Image Processing*, vol. 7, no. 2, pp. 259 – 265, 1978.

[33] C. Glasbey, "An analysis of histogram-based thresholding algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 6, pp. 532 – 537, 1993.

[34] J. M. S. Prewitt and M. L. Mendelsohn, "The analysis of cell images*," *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035–1053, 1966.

[35] G. W. Zack, W. E. Rogers, and S. A. Latt, "Automatic measurement of sister chromatid exchange frequency." *Journal of Histochemistry & Cytochemistry*, vol. 25, no. 7, pp. 741–753, 1977, pMID: 70454.

[36] T. W. Ridler, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 8, pp. 630–632, Aug 1978.

[37] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, jan 1979.

[38] J. Kapur, P. Sahoo, and A. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985, cited By 1638.

[39] J.-C. Yen, F.-J. Chang, and S. Chang, "A new criterion for automatic multilevel thresholding," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370–378, Mar 1995.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

Advanced web analytics tool for mouse tracking and real-time data processing

Lukáš Čegan[1], Petr Filip[2]
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
[1]lukas.cegan@upce.cz
[2]petr.filip1@upce.cz

***Abstract*—Web analytic tools offer important support for better recognition of the web user's behavior, identification of bottlenecks and errors in user interface design, performance measurement of web environment, monitoring of website availability or recommendation of appropriate website content. These tools are based on tracking techniques and sophisticated algorithms that process and evaluate large volumes of captured data. In this paper, we propose a new solution to capture mouse movements of web users, to identify their area of interest. This solution is based on real-time data transformation, which converts discrete position data with high sample period to predefined functions. This transformation has a high degree of accuracy, which is exemplified by case scenarios. The result of this solution is a significant saving in data, transmitted from the client to the server, which leads to significant savings in system resources on the server side.**

Keywords—mouse tracking; real-time data processing; user behavior

I. Introduction

The World Wide Web is definitely the most widespread service of the current Internet with more than 3.5 billion users. Most of the population of the developed world uses this service daily for shopping, communication, information search, company presentation, advertising, etc. Many companies are trying to leverage this fact and try to commercialize their products and services in this large market. However, this creates considerable competition among companies, which is reflected in the high demands on the quality of web applications. For these reasons, increasing emphasis is placed on the quality of user experience design in the process of developing web applications. This better user experience design leads to greater user satisfaction with the web application and hence greater revenues. The quality of user experience design can be measured and evaluated using appropriate tools to track the activities of each web user. Fortunately, today there are enough tracking tools on the market. The range of tracking functions of these tools is different, as well as the method of data communication between the web client and the tracking server. In many cases, the volume of data transferred is considerable, which has a negative impact on the quality of the service provided. One of the important tracking functions for evaluating user experience is to track mouse movement. Unfortunately, this feature is one of those which is based on logging a large amount of data that is necessary to send to the server for evaluation. This problem can be addressed by the new solution that is presented in this paper. This solution is based on tracking mouse movement and subsequent vectoring of the mouse path on the client side, resulting in significant reduction of transmitted data while maintaining accuracy of measurement. The quality of the vectoring function is judged in the case study that is part of this paper.

The structure of this paper presents as follows. After introducing the objective of this paper, the related work is presented in Section II. Next, in Section III, a proposed solution for tracking mouse movements of web users is described. The concept of mouse movement tracks vectorization is given in Section IV. The Section V described the performance evaluation of proposed solution. In Section VI is mentioned experimental results and discussion covering results of the presented work. Finally, the last Section VII gives conclusions and future research opportunities.

II. Related Work

Mouse tracking is a basic technique which produces a large amount of user behavioral data that can be used to evaluate and improve a wide range of areas such as user experience design, prefetching and recommendation systems, and psychological studies. A number of research papers dealing with the monitoring of activities and tracking mouse movements of web users has been published by various authors in recent years. Many of these research studies focus on evaluating web user behavior for recommendation and prediction modeling in the e-commerce area [1], [2], [3]. Additional published papers are devoted to discovering the relationship between the users' eye movement and mouse movement [4], [5]. The area of neural and psychological studies cannot be omitted either [6], [7], [8]. Unfortunately, less scientific work is devoted to the motion capture tool itself [9], [10]. In addition to solutions published in scientific work, commercial tools are available on the market. These tools usually offer a comprehensive set of features to monitor web users. They also have analytic capabilities to evaluate captured data whose outputs provide valuable information to business analysts about website performance.

978-1-5386-0889-0/17/$31.00 ©2017 IEEE

The main tools are Crazy Egg, Clicktale, Hotjar, Mouseflow, SessionCam, LuckyOrange, UsabilityTools, Inspectlet and Hoverowl. The operation of commercial and academic web tracking tools is based on the same principle, namely capturing defined events by JavaScript API that occur during a web user session. This captured data is periodically asynchronously sent to the logging server, where they are processed and stored for future use. However, this communication is very demanding in terms of the amount of data that needs to be transferred from the client side to the server side. The proposed mouse pathway vectorization, as described in this paper, largely addresses this problem.

III. Proposal Solution

A proposed solution is based on a new advanced web analytics platform, which provides a range of functions through which it is possible to tracking and analyze mouse movements in real-time with a small amount of data transmitted to the server. The architecture of the proposed platform is shown in Fig. 1.

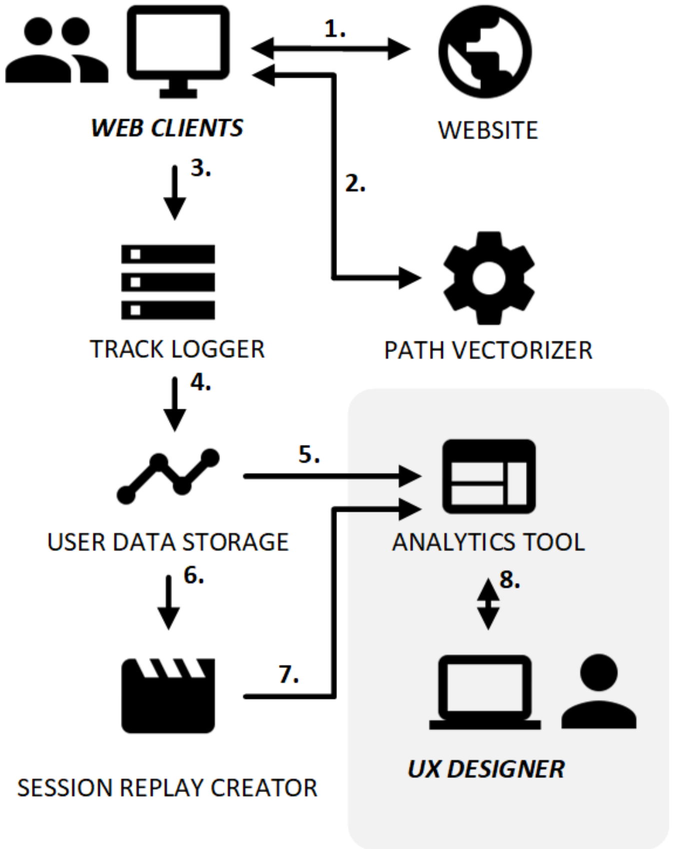


Fig. 1. Architecture of tracking and analytics platform. (1) Web client requests website from web server and (2) vectorizer library from path vectorizer (also web server). User activity is tracked and data is sent asynchronously to Track logger. (4) processed data is stored in User data storrage. (6) Seesion replay creator use data for video. Analytical tool use data and video for user report. (8) UX designer use analytical reports.

The proposed platform consists of five parts: tracking logger, track vectorizer, user sessions storage, session replay creator and analytics tool.

A. Track logger

The track logger is a component responsible for data transfer from client to server. The component is implemented as a JavaScript library that sends data to the server using asynchronous POST HTTP requests. Before sending an HTTP request, the data is processed by a vectorization component. To minimize the transmitted data, they are also compressed using the LZ77 algorithm (Lempel-Ziv, 1977). This is used by the LZ-UTF8 library [11].

B. Path vectorizer

Path vectorizer is a component responsible for the mouse movement path vectorization. This vectorization minimizes the data that is essential for describing the mouse trajectory. A description of the internal functioning of this component is described in detail in Section IV.

C. User data storage

User data storage is a component, providing storage of measured user session data. As a data repository, relational databases Oracle is used. The stored data is used by components: Session replay creator and Analytics tool.

D. Session replay creator

Session replay creator is a component that creates a preview of each user session. The preview displays mouse movement and each visitor activity in a time. At this time, it is not possible to create a video in the web environment that would record the user's work in the client's web browser. For this reason, creating a preview is based on capturing all HTTP communication between the client and the server. These web resources are retrieved and stored by the logging server. From these sources, a user session preview is created on request and played in the built-in iFrame, where the user's activities are simulated. All sources of each HTML page must be archived so that the user session can be previewed, even after the source page is updated. Unfortunately, it is not possible to create a preview of a session when a third-party site redirection occurs, such as a logging process with Google's oAuth service.

E. Analytics tool

Analytics tool manages measured user sessions. Each session contains metadata that allows users to search and filter sessions. The user sessions can be played on the built-in player. For more convenient analysts' work, a preview can be played back quickly. At this point, no other logic is implemented in the tool that could segment users by other criteria such as a conversion ratio.

The next Figure 2. illustrates a sequence diagram that describes the integration of each component of the proposed solution, from time perspective. The communication starts, when the Web browser sends a HTTP request to the web server. The web server processes a request and then sends a response (web page). Next, the web browser parses the source code of the web page, and it starts loading the tracking and vectorization library. After the web browser receives the library, the monitoring and evaluation of user activity is initiated. The catchEvent feature ensures that both the mouse movement data, and the page scrolling and resizing data, are captured and stored in the FIFO queue. From this queue, the data is sequentially removed and processed using a vectoring tool. Then, the processed data is sent asynchronously to the track logger, which uploads it in the storage.

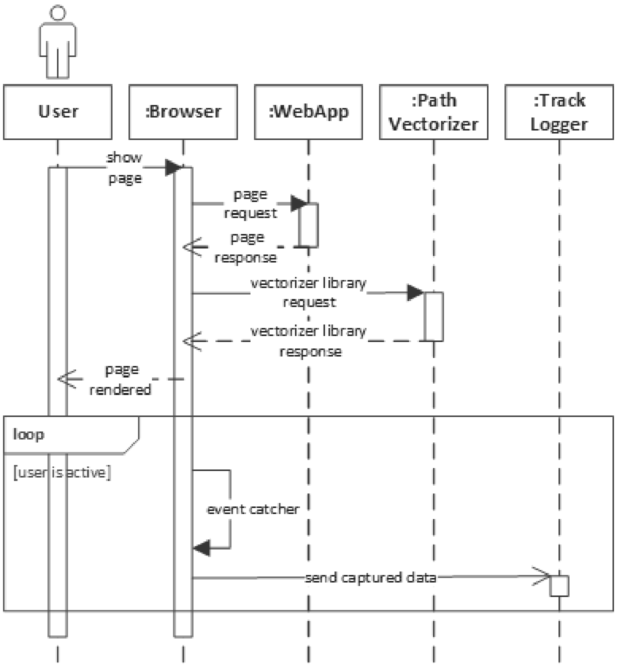


Fig. 2. UML sequence diagram of the mouse tracking implementation

IV. Path Vectorizer

The path vectorizer is a core component of a proposal analytical tool. The component is responsible for evaluating the captured position data and their subsequent vectorization. Evaluation is done in real time. The vectorization process is illustrated in the following Figure 3.

The vectorization process starts with an event, when a web page is rendered in the client browser. Subsequently, the tracking module is initialized, which waits until the occurrence of the Web API event “mouseMove”. This event fires a function that saves data to a local storage (FIFO queue). Furthermore, the distance between the last and the penultimate point of the queue, and their time difference, is calculated. In the case of a small positional distance or a large time difference, the motion of the mouse is evaluated as interrupted, and therefore the timer is not reset. In other cases, the timer is reset. The timer is an element that stores the time stamp of the last known mouse movement activity. When the timer is inactive for three seconds, an event occurs that initiates the vectorization process of the stored data. First, all the data is removed from the queue. These data then enter into the vectoring calculation. The result of the calculation is the curve expressing the mouse path and a time curve expressing the movement of the cursor over the curve, over time. Once the boundary volume of data is collected, these curves are sent to the server.

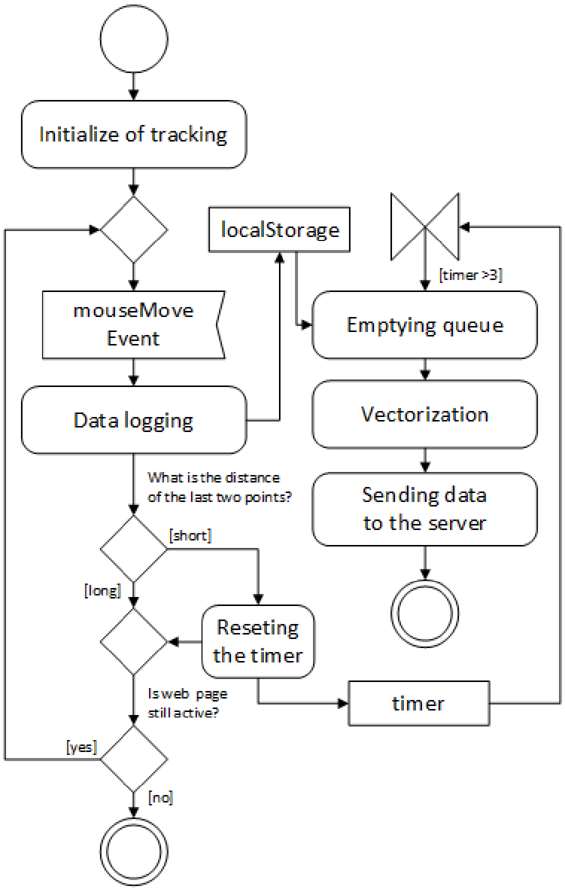


Fig. 3. UML activity diagram of JavaScritp tracker

The Vectorization activity, shown in the diagram above, consists of a sequence of atomic tasks:

- finding boundary points,
- and replacing points with a curve.

Boundary points are defined as points of the dataset where the curve type is changed or where the cursor movement speed is changed. The proposed solution distinguishes only two types of curves: line and b-spline. Finding boundary points is based on analyzing the sequence of data points and finding a breakpoint.

V. Performance Evaluation

The effectiveness of the proposed solution has been investigated in an experimental test. For the needs of the experiment, a special test website was created. The test website has been deployed to a testbed platform. The platform consists of the physical machine Dell E6440, Intel i5 (4310M), 2.70 GHz, 8GB RAM, Windows 10 64 bit. and the virtualization platform Oracle VirtualBox 5.1.26. The virtual machine host represents the desktop client with web browser Chrome 60. The guest represents the server side with operation system Debian 9.1 and the web server Apache 2.4.25.

In order to simulate a more realistic network environment, test was performed with several different network parameters (if allowed). Individual measurements were made in three different simulated network environments: (A) 3G – 1 Mbit/s bandwidth and 300ms latency, (B) LTE – 10 Mbit/s and 50ms latency and (C) Fiber – unlimited Mbit/s bandwidth and 50ms latency. For creating a simulation environment Linux tool Netem (Network Emulator) was used which provides functionality for variable delay, loss, duplication and re-ordering with combination of traffic shaper tool TBF (Token Bucket Filter), which allows the slowing down of transmitted traffic, to the specified rate (Fig. 4).

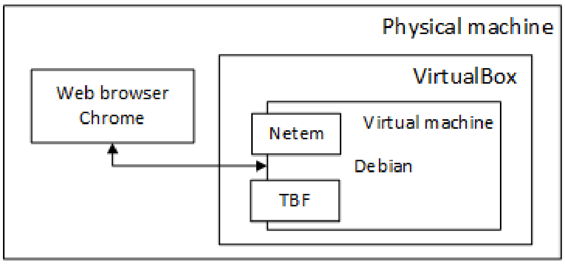


Fig. 4. Communication schema of testbed platform

The impact of the vectorization algorithm was conducted on three test scenarios. The test scenarios included captured mouse movement data:

A. without vectorization,
B. with vectorization and without using compression,
C. with vectorization and with compression.

In order to compare the values of each scenario, all three scenarios were executed at the same time. Experimental measurements were performed on fifty user sessions.

The following figure demonstrates the user's mouse movements on one page (Fig. 5). All the points that were captured using the web and the "mousemove" event are visible on the chart – Scenario A.

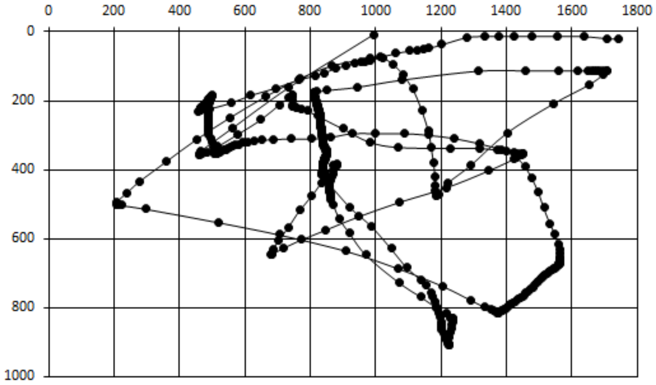


Fig. 5. Graph of mouse movement

Figure 6. shows the same mouse movements as in the previous figure, but points were processed using a vectoring tool. As can be seen in the figure, the number of points expressing the mouse's path has significantly decreased with preserved position accuracy – Scenario B.

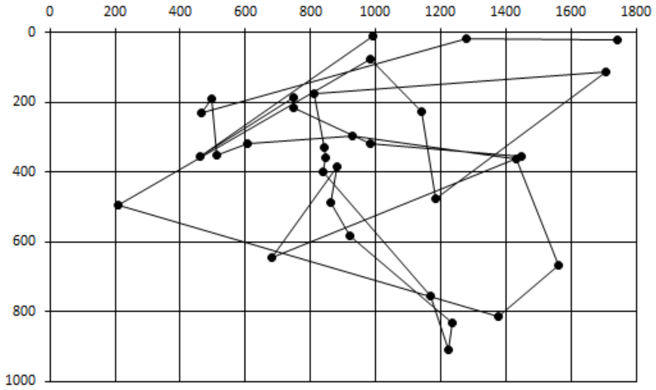


Fig. 6. Graph of mouse movement

VI. Experimental Results

Table I. showed cumulative values of HTTP communication of all test user sessions. For each scenario, the value of the number of HTTP requests is expressed. Only requests that sent the captured data to the server, are counted. The last column of the table shows the total volume of data transferred for all requests.

TABLE I. Cumulative Communication Values

Scenario	Requests	Data size [kB]
A	2 082	13 824
B	285	3 125
C	239	2 193

Table II. shows a result of the vectorization process. In the first column, the type of curve is listed. For each curve, the number of points that have been replaced, their total length, the total cursor time on the curve, and the accuracy of the substitution are calculated. Calculation of replacement accuracy is based on the residual sum of squares.

TABLE II. Vectorization

Curve	Points	Length [px K]	Time [s]	Accuracy[RSS]
Line	21 391	20 862	5 220	0,87
B-spline	2 118	1 237	318	0,69

VII. Conclusion

This paper presents a new solution for tracking web user activity that allows understanding of how a user interacts with websites. This interaction includes path movement, scrolling, and clicks. The main part of the proposed solution is an advanced component for vectorization of mouse movement trajectory that dramatically reduces the amount of data that needs to be transferred to the server. The degree of vectorization accuracy has been investigated and evaluated by the performance experiment. As can be seen from the results, the vectoring of the captured points, results in considerable savings of transmitted data while maintaining a high degree of accuracy.

As future work, we are planning to implement the new module to a session replay creator component which allows creating of a full-featured video file for more accurate reconstruction and playback of visitor activities. Combining this tool with other analytic tools, such as Google Analytics, provides a full picture of web user behavior.

ACKNOWLEDGMENT

This work is published thanks to the financial support Faculty of Electrical Engineering and Informatics, University of Pardubice under grant SGS_2017_025 “Active monitoring of web users’ behavior”.

REFERENCES

[1] Md. Tanjim-Al-Akib, Lutfullahil Kabir Ashik, Hosne-Al-Walid, Krishanu Chowdhury, “User-modeling and recommendation based on mouse-tracking for e-commerce websites,” in *Computer and Information Technology (ICCIT)*, 2016, pp. 517-523.

[2] Song HengJie, Liao Ruoxue, Zhang Xiangliang, Miao Chunyan, Yang Qiang, “A mouse-trajectory based model for predicting query-url relevance,” in *Proceedings of the National Conference on Artificial Intelligence*, 2012, pp. 143-149.

[3] Arapakis Ioannis, Leiva A. Luis, “Predicting User Engagement with Direct Displays Using Mouse Cursor Information,” in *Proceeding of the 39th International ACM SIGIR conference*, 2016, pp. 1-10.

[4] K. Rodden, X. Fu, A. Aula, I. Spiro, “Eye-mouse coordination patterns on web search results pages”, in *CHI O8 extended abstracts on Human factors in computing systems*, 2008, pp. 2997-3002

[5] Q. Guo, E. Agichtein, “Towards predicting web searcher gaze position from mouse movements” in *Proceeding of 28th international conference on Human factors in computing systems*, 2010, pp. 3601-3606

[6] Eric Hehman, Ryan M. Stolier, Jonathan B. Freeman, “Advanced mouse-tracking analytic techniques for enhancing psychological science” in *Group Processes & Intergroup Relations*, Vol 18, Issue 3, 2015, pp. 384 – 401.

[7] Thomas A. Farmer, Sarah A. Cargill, Nicholas C. Hindy, Rick Dale, Michael J. Spivey, “Tracking the Continuity of Language Comprehension: Computer Mouse Trajectories Suggest Parallel Syntactic Processing” in *Cognitive science*,Volume 31, Issue 5, 2007, pp. 889–909

[8] Sara Incera, Theresa A. Markis, Conor T. McLennan, “Mouse-Tracking Reveals When the Stroop Effect Happens.” in *The Ohio Psychologist*, 2013, pp. 32-34.

[9] L. A. L. Torres and R. V. Hernando, “(smt) real time mouse tracking registration and vizualization tool for usability evaluation on websites” in Proceedings of the IADIS International Conference on WWW/Internet, 2007, pp. 187-192

[10] Clements Schefels, Sven Eschenberg, Christian Schöneberger, “Behavioral Analysis of Registered Web Site Visitors with Help of Mouse Tracking” in IEEE 14th International Conference on Commerce and Enterprise Computing, 2012, pp.33-40

[11] GitHub (2017, Sep. 2) A high-performance Javascript string compression library. [Online]. Available: https://github.com/rotemdan/-lzutf8.js

AUTHOR INDEX