INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

RELATIONSHIPS BETWEEN THE CONCEPTS IN THE DESIGN ONTOLOGY

Mario Štorga[1], Dorian Marjanović[1] and Mogens Myrup Andreasen[2]

[1] Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia
[2] Department of Mechanical Engineering, Technical University of Denmark

ABSTRACT

Presented research is aimed to the investigation of the nature, building and practical role of a *Design Ontology* as a potential formal description of the shared engineering knowledge in design domain. As the part of the research results, this article summarizes our experience of identification and formal characterization of the large number of relations that can be extracted from the domain of product/design description. Based upon this formalization authors illustrate the possibilities for creating a more definite formal design model.

Keywords: formal design model, product modelling, relationships, design ontology, complexity management

1 INTRODUCTION

It is recognized for a while that insight into engineering knowledge is one of an enterprise's most important assets, decisively influencing its competitiveness. To describe the complex pattern of the engineering knowledge content, different theoretical and practical design models often contain a vast amount of diverse elements linked together in a variety of mostly undocumented networks without precise explanation of the associations meaning. Such situation motivated the research aimed to the investigation of the nature, building and practical role of a *Design Ontology* [1] as a potential formal description of the shared engineering knowledge in design domain. As the part of the research results, this article presents our findings regarding the identification and formal characterization of the relations diversity that can be extracted from the domain of product/design description.

2 RESEARCH METHOD

Any domain with a determinate subject matter has its own terminology, a distinctive vocabulary that is used when talking about characteristic concepts that compromise the domain. But the domain space is not revealed in its corresponding vocabulary only. In order to form the logically correct statements about a situation in a domain, rules and restrictions (often called domain axioms) governing the way terms in vocabulary should be utilized, must be provided and clarified. The role of axioms is to constrain the meaning of the terms in vocabulary sufficient to enable consistent interpretation of statements based on the vocabulary. Only with this additional information available, it is possible to understand both the nature of the individual concepts that exist in the domain and the associations they bear to one another [1]. Mixed approaches of existing methodologies [2] have been aimed in research presented in this article to the successful formalization of a *Design Ontology*, with the main goal of building the vocabulary of design domain and specification of main terms meaning. Specification of the terms' meaning has presumed definitions and indications of how domain concepts are inter-related, which collectively impose a structure of the domain informational model and constrain the possible interpretations of vocabulary terms.

In order to formalize the meaning of the different relationships that exists between concepts in a design domain, building the general relations taxonomy considering their nature was one of the first research steps. In order to make formal characterization of the numerous relations extracted from the *Design Ontology* background theories, relationships for design description domain were classified and defined by axioms considering their logical properties as reflexivity, irreflexivity, transitivity, symmetry, asymmetry, antisymmetry. Such proposed formal design model was implemented using the OntoEdit® ontology development environment and the instances of the *Design Ontology* elements were created based on test examples. The formal design model was then tested for formal correctness using the automated reasoning mechanism of the ontology development environment, and based on the test results final conclusions were drawn.

3 RELATIONSHIPS MODELLING

During the *Design Ontology* terms extraction phase, the hundreds of definitions for the terms were derived based on a high level ontology (SUMO by IEEE - www.ieee.org) and background theories, defining the ambiguous meaning for every single domain concept [1]. Such definitions have been formalized using the extracted relationships between concepts that exists in design domain e.g. *is result of, follows, describes, are kind of, have an input, etc*. Most of the extracted relationships have no explanation of their nature and meaning in the background theories, and are often described as a causal relation, with the purpose to denote their existence, without further explanation of their logical behaviour. Therefore, at this research stage, autjors decided to take a further insight into the research of the relationships that exist in the different design and informational models in order to create relations taxonomy as an important structural part of *Design Ontology*.

Relationships in the design models

In engineering design domain, Hubka and Eder [3] define relationship as the real or meaningful dependence or interaction between two or more objects or phenomena of abstract or concrete kind. They also conclude that objective, exact and describable relationships of the natural science are important also for the design science. According to their contribution the main type of the relationships following the proposals from the natural sciences are:

- Similarity: relationship between two systems based on certain characteristics and/or properties that they have in common.
- Analogy: co-incidence of important characteristics and properties belonging to objects of phenomena
- Homomorphy: relationship between two systems in which every component and every relationship in one system permits conclusions about the elements and relationships in the second system, but conversely conclusions about first system cannot be derived from the second.
- Isomorphy: relationship between two systems in which every component and every relationship in one system can be uniquely mapped to a component or relationships of the other system, and the converse mapping also holds.
- Equivalence: objects are termed equivalent if an equivalence (reflexivity, symmetry, transitivity) relationship exists between them.
- Identity: relationship between objects, states, statements, etc. in which there is complete coincidence of some characteristics between them.
- Mathematical functions: functional relationships between objects, which may be modelled (its complexity) mathematically.
- Causality: asymmetric relationship between a cause and its consequence.
- Couplings: occur when certain outputs of a system are simultaneously inputs to another system.
- Goal-means: asymmetric relationship between a system of goals, and the system of means by which the goals can be realized.
- Spatial: describes the arrangement of elements, the one in a relation to the other.
- Logical: relationships of logical nature which may be modelled by Boolean algebra.
- Time: describes the arrangement and progression of processes, occurrences, happenings, etc. along the axis of elapsing time.

A numerous researches have continued upon this approach and reused this classification in different design modelling methods, without deeper reasoning about relationships' nature.
In a literature a few approaches (like Eppinger et al. [4]) could be found that are considering the association between decomposed elements in the design domains as a pattern of interactions. Mapping

of the patterns of interactions in product architecture, development processes and organizations is accomplished using matrix based methods, reducing in this way the complexity of the model into simple structures denoting the interactions between components, working tasks, and peoples involved. It seems that the matrix methods are also based upon classification without deeper reasoning about the nature of the relationships shown in matrices.
Pavković et al. [5] in their work distinguish the relations' semantics in two contexts:

- Context of the general object-oriented information modelling: dependency, generalization, association and realization.
- Context of the product and design process modelling: dependency, affiliation, sequence, responsibility, hierarchy, constraints and conditions.

The important fact of such approach is that design process and product are not viewed as a static institutionalised structure, but rather as a dynamic network that is constructed in real time as development project and design knowledge evolves. In such dynamic pattern the designers have to establish proper coordination between different structures and relations between them [6].
In the one of the latest discussion about relationships in product structures, McKay et al. [7] concluded about three groups of the relationships in product structures:

- Relationships needed to describe a product at stage of life-cycle and time: composition, constitution, inherence, qualification and quantification, designation.
- Relationships needed to support configuration management: equivalence, alternation, variation, order, and transformation.
- Relationships needed to support entity realisation: articulation, factorization, consolidation.

McKay et al. [7] claim that each of the previous mentioned relationship groups contains a limited number of type of relationships and governs the extent to which a given information system can support product life-cycle processes to which the product will be subjected.

Relationships in the informational models

Since the mid-1970's it has become clear that the calculus of relations is a fundamental conceptual and methodological tool in computer science just as much as logic. While computer science applications are evolving rapidly in several areas like communication, programming, software, data or knowledge engineering, exact sciences are needed to understand existing methods. It has become more and more suitable to use formal approaches to handle design-, algorithmic- or information complexity. Among several formal approaches, relation algebra [8] has been used as a basis for analysing, modelling or resolving several computer science problems such as: program specification, heuristic approaches for program derivations, database and software decomposition, program fault tolerance, testing, data abstraction, information coding and spatial reasoning. While axiomatically simple, relation algebra has proved to be able to cover a large variety of information structures.

Authors in design science research field in their discussion about requirements for a technical memory for the engineering knowledge structuring (Mekhilef et al. [9]) has proposed the typology of the necessary informational relations following the object oriented paradigm:

- Vertical relations, i.e. links of specialization and instantiation: concept-object instantiation, object-instance instantiation, concept-concept inheritance, object-object inheritance.
- Horizontal relations, i.e. property links, attribute and correlation: object-attribute (characterization), object-object and concept-concept (association).

In recent approaches to construction of domain ontologies, relations are typically considered as the general associations which can be shared by distinct pairs (triples, etc.) of domain's individuals [10]. The traditional goal of an ontological inquiry in particular is to divide the domain of discourse "at its joints", it means to discover fundamental categories, or kinds, into which the domain's concepts naturally belong. Thus, relationships are identified by abstracting particular features of individuals and

are often characterized as being of a higher (more abstract) logical type than the individuals that represent them.
The presented analysis has drawn out important key points regarding the relationships as the constitutional elements of the informational models:

- Relations are the key elements of the formal model that enables the design rationale patterns for capturing the common elements of design reasoning structures in a form favourable to trace and reuse.
- Defining the logical characterization of the relations by formal rules derived on their natural properties, is enabling the effective automated reasoning about described situation in domain of discourse.

4 THE DESIGN ONTOLOGY RELATIONSHIPS

The IDEF5 (www.idef.com, Knowledge Base Systems 1994) was chosen in presented research as the ontology description capture methodology since it provides theoretically and empirically well-grounded methods specifically designed to assist in creating, modifying, and maintaining ontologies for different areas. Following this approach, and results of the relationships in informational and design models analyse, authors have succeeded to around fifty different relations extracted in this phase of *Design Ontology* building has been categorised into seven main groups, namely: compositional, spatial, role, dependency, influence, temporal and general relations (Figure 1).

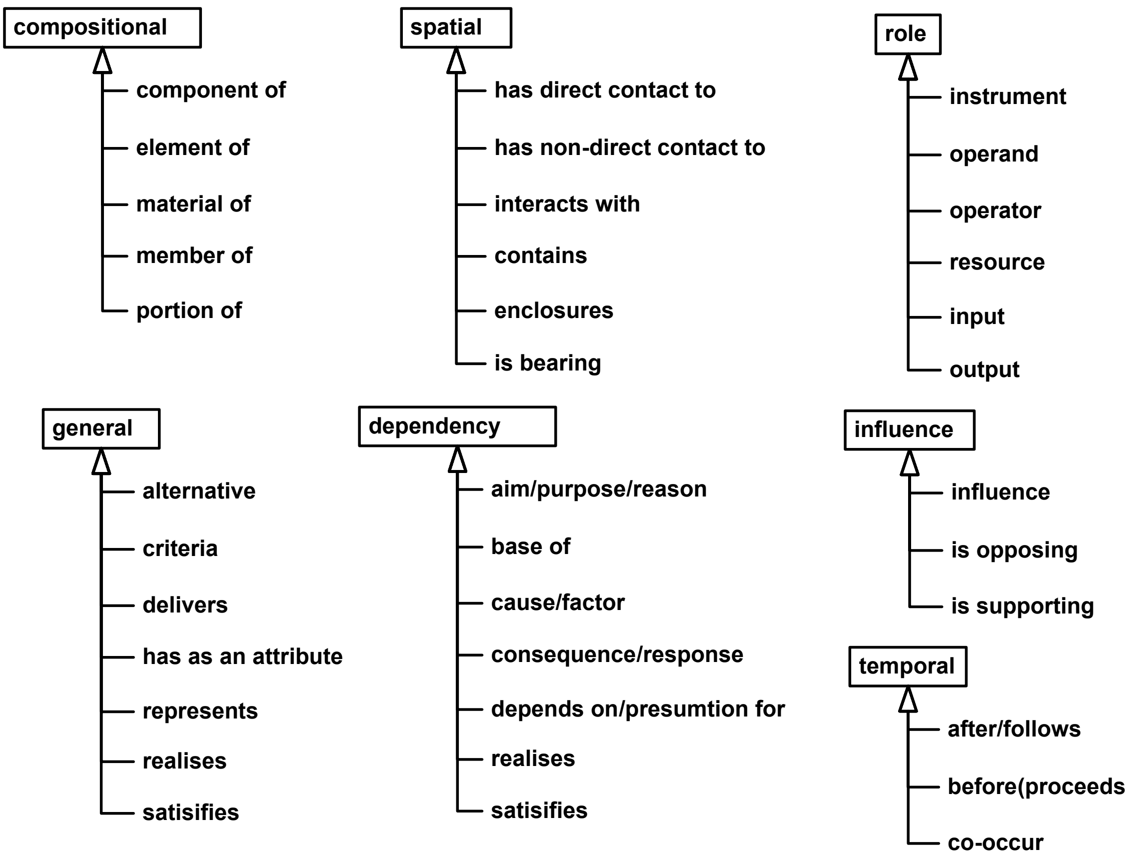

Figure 1. Taxonomy of the Relations in Design Ontology

Compositional relations

The *Compositional relations* are a kind of relations that capture semantics of whole/part concepts based on the logical theory of structures called mereology [11]. The *Compositional relations* are antisymmetric, irreflexive and transitive by their nature. The main terms extracted from GDMS and classified as a *Compositional relations* are:

- *component of* – is used to relate a complex physical entity and its physical components (organ/transformational organism, engineering component/assembly, assembly/device, etc.)
- *element of* – is used to denote that an simple entity is functional element of complex entity (fact/information, assembly/product structure, operation/operational chain, element/set, etc.)
- *material of* – is used to describe that one entity is partly made of some material (material/engineering component, etc.)
- *member of* – describes the fact that an entity is a member of some collection (product/product family, product/assortment, human/group, etc.)
- *portion of* – describes the relationships between two entities, one being included in the other (constant quantity/functional quantity, etc.)

Spatial relation

The *Spatial relations* are a kind of relations that capture semantics of the geometric, physical and other form of connections, contacts or interactions between physical entities. All *Spatial relations* are irreflexive. Some of them are transitive (for example contain) and others are symmetric (for example direct contact). The main terms classified as *Spatial relations* are:

- *has direct contact to*– is used to describe that two entities are in physical contact (engineering component/engineering component, organ/organ, surface/surface, etc.)
- *has non-direct contact to* – is used to describe that two entities are not in physical contact but they are components of the same complex entity (engineering component/engineering component, organ/organ, surface/surface, etc)
- *interacts with* – is used to denote that two entities perform some action one to another (organ/organ, surface/surface, etc.)
- *contains* – is used to describe that one entity are taking the space occupied by other entity (material entity/material entity, etc.)
- *enclosures* – synonym of contain
- *is bearing* – describes the spatial arrangement of the two material entities in observed plan (material entity/material entity, etc. – could be further classified as left-of, above, behind, inside, etc.)

Role relations

The *Role relations* are the kind of relations relating the physically distinguished roles of the different elements of the process. The *Role relations* are antisymmetric, irreflexive and intransitive. *Role relations* include, for example, the agent, patient or destination of a transformation that take a place during the particular process. The main terms classified as a *Role relations* are:

- *instrument* – is used to denote that an entity is a tool for creating transformations in a specific technical process (object/process, etc.)
- *operand* – is used to denote that an entity is object of transformations in a specific technical process (information/technical process, matter/technical process, energy/technical process, etc.)
- *operator* – is used to denote that an entity is an active creator of a transformation in a specific technical process by exerting the effects that drive and guide the process (management system/transformational system, executive system/transformational system, informational system/transformational system, etc.)
- *resource* – is used to describe that an entity is necessary present at the beginning of the process, used during the process, and could be changed in a process (energy/process, object/process, etc.)
- *input* – describes the state of the operand at the beginning of the process
- *output* – describes the state of the operand at the end of the process

Dependency relation

The *Dependency relations* are kind of relations that capture semantics of the fact that one entity in domain depends existentially on another entity. The *Dependency relations* are antisymmetric, irreflexive and transitive. The main terms classified as a *Dependency relations* are:

- *aim* – is used to denote that an entity is an intended (planned) purpose or is a reason for existence of another entity (specification/design attribute, etc.)
- *base of* – is used to denote the entity from which another entity is derived, or based on (fact/argument, etc.)
- *cause* – is used to describe that an entity somehow causes progress, activity or existence of another entity (design property/design characteristic, etc.)
- *consequence* – is used to denote that an entity is a product, result, or response on existence, activity or work of another entity (need/problem, transformation/need, etc.)
- *depends on* – is used to describe that an entity existentially depends on another entity (compositional characteristic/organ, etc.)
- *factor* - synonym of cause
- *presumption for* - synonym of depend on
- *purpose* - synonym of aim
- *reason* - synonym of aim
- *response* - synonym of consequence
- *result* - synonym of consequence
- *stimuli* - synonym of cause

Influence relations

The *Influence relations* are kind of relations that capture semantics of the fact that one entity has some effect or impact on another concept. The *Influence relations* are antisymmetric, irreflexive and transitive. The main terms classified as an *Influence relations* are:

- *influence* – is used to denote that an entity has influence on progress, activity, or existence of another entity (life cycle meeting/relational properties, etc.)
- *is opposing* – is used to denote that an entity challenge correctness of another entity (argument/entity, etc.)
- *is supporting* - is used to denote that an entity support correctness of another entity (argument/entity, etc.)

Temporal relations

The *Temporal relations* are kind of relations that capture semantic of the time-depending relations between entities, based on the temporal logic and main concepts as time interval and moment. The *Temporal relations* are antisymmetric, irreflexive and transitive. The main terms classified as a *Temporal relations* are:

- *after* – describes that the time interval of activity for an entity starting latter on a time progression line than ending time interval of activity for another entity (process/process, function/function, etc.)
- *before* - describes that the time interval of activity for an entity ending before on a time progression line then starting the time interval of activity for another entity (process/process, function/function, etc.)
- *co-occur* – describes that an entity exists or is active in the same time interval as another entity (process/process, function/function, etc.)
- *follows* - synonym of after
- *proceeds* - synonym of before

General relations

The *General relations* are kind of relations that capture semantic of very general predicates, and therefore were not possible to characterize them into one of previously outlined groups. They are all antisymmetric and irreflexive. Some of them are transitive (for example describe, realise). The main terms classified as a *General relations* are:

- *alternative* – is used to denote that an entity could take a place of another entity (engineering component/engineering component, concept/concept, etc.)

- *criteria* – is used to denote that one entity is criteria for evaluation of another entity (need/concept, etc.)
- *delivers* – is used to describe that an entity by its activity delivers another entity (executive system/effect, etc.)
- *describes* – is used to denote that an entity somehow indicate, express, picture, represent, describe another entity (design property/behaviour, design characteristic/constitution, function/activity, etc.
- *has as an attribute* – is used to describe that an entity is characterized by another entity (engineering component/task, society system/problems, etc.)
- *represents* - synonym of describe
- *realises* – is used to describe that an entity physically realises another entity (organ structure/working principle, assembly/organ, effect/technical process, etc.)
- *satisfies* – is used to denote that one entity fulfils some requirement or expectation (product/need, etc.)

It should be clear that presented classification of the main groups of relation is not definite and is based upon very narrowly defined background [1]. A different design models include a different associations between its elements that could be also easily included in this classification. Besides, almost every of proposed relation could be further specialized. In this context, presented approach is intended to guide the implementation and deployment of proposed formal language in particular design situation, where will be extended with additional terms for specific purpose.

5 FORMAL MODEL VALIDATION

The validation of the proposed formal model by instantiation of the main terms and relations enabled the test pattern favourable for the knowledge model consistency checking. The implementation began with building the knowledge tree by mapping every particular term from the *Design Ontolog*y to the OntoEdit® (ontology development environment by Ontoprise GmbH – for more details see www.ontoprise.de) concept hierarchy tree and defining the relations in a manner of OntoEdit® relational axioms. The nature of the extracted relations is defined by adding the additional axioms describing the logical properties of every relation's group. Implemented relation axioms together with the automated reasoning mechanism of the ontology development environment enabled us to locate the proposed formal model's logical errors locating and correcting. Unformal questions used during the terms and relations extraction, have been formalized into semantic queries applied on the test instances set [1], in order to prove the applicability of the proposed *Design Ontology* for the capturing, storing, querying and reusing the engineering knowledge that evolves during the product development sequence.

6 CONCLUSION

In order to support the effective implementation of engineering knowledge management, presented research is looking for the comprehensive *Design Ontology* as an instrument for achievement the full interoperability between different participants (humans and computer systems) of development processes. In order to overcome the shortcomings of different research approaches, which orientation has led to assumption that all what need to be said about a situation in a domain of discourse can be said without appeal to the formal aspects of the associations that exists between domain concepts, our focus were on explanation of the nature of relations in design description domain. With this goal authors have proposed taxonomy of relations in design description domain as an integral part of the entities describing the abstract part of *Design Ontology* [1]. The proposed classification and relation definition enables developing the formal language for articulation of the different viewpoints on design, design genesis, and design rationale. Future work will focus on developing the advanced tools to assist in the knowledge management processes during the product development based on ontology proposal.

ACKNOWLEDGMENTS

This research is part of funded project "Models and methods of improving the computer aided product development" supported by the Ministry of Science, Education and Sport of the Republic of Croatia.

The presented paper is result of the research performed in co-operation with Section of Engineering Design and Product Development, Technical University of Denmark.

REFERENCES

[1] Štorga, M., Andreasen, M.M. and Marjanović, D., "Towards a formal design model based on a genetic design model system", Proceedings of the 15th International Conference on Engineering Design ICED 05 Melbourne - Australia, Samuel, A. and Lewis, W. (editors), Engineers Institution of Engineers, Australia, II National Circuit, Barton, Melbourne, 2005.

[2] Ahmed, S., Kim, S., Wallace, K.: "A methodology for creating ontologies for engineering design"; Proceedings of DTM 2005; ASME 2005 Design, Theory and Methodology Conference; Long Beach, California, USA; 2005.

[3] Hubka, V. and Eder, W.E., "Theory of technical systems – a total concept theory for engineering design", Springer-Verlag, Berlin Heidelberg, 1988.

[4] Eppinger, S.D., Salminen, V.: "Patterns of product development interactions"; Proceedings of the 13th International Conference on Engineering Design ICED 01 Glasgow - UK, (editors), The Design Society, 2001.

[5] Pavković, N., Bojčetić, N., Marjanović. D.: "Modelling the relation network in the integrated product and design process model"; Proceedings of the 7th International Design Conference – DESIGN 2002, Dubrovnik, Croatia; Marjanović, D. (editor), University of Zagreb, The Design Society, pp. 391-396, 2002.

[6] Duffy, A.H.B., Andreasen, M.M., O'Donnell, F.J.: "Design Co-ordination", International Conference on Engineering Design (ICED'99), Munich, Germany, 24 - 26 August 1999.

[7] McKay, A., Hagger, N.H., Dement, C.W., de Pennington, A. and Simons, P., "Relationships in product structures", Proceedings of the 7th Workshop on Product Structuring – Product Platform Development, Chalmers University of Technology, Goteborg, 2004.

[8] Maddux, Roger D.: "Relation Algebras", vol. 150, Studies in Logic and the Foundations of Mathematics. Elsevier Science, 2006.

[9] Mekhilef, Bourey, Bigand, "An UML modelling of an architecture for knowledge documentation", Proceedings of the 14th International Conference on Engineering Design ICED 03 Stockholm - Sweden, (editors), The Design Society, 2003.

[10] Knowledge Based Systems, Inc.: "IDEF5 Method Report"; 1994.

[11] Salustri, F.A., Lockledge, J.C.: "Towards a formal theory of products including mereology"; Proceedings of the International Conference on Engineering Design ICED 99, Munich, Germany, 1999.

Contact: Mario Štorga
University of Zagreb/Faculty of Mechanical Engineering and Naval Architecture
Chair of Design and Product Development
Ivana Lucica 5
10000, Zagreb
Croatia
Phone: +385 1 6168 119
Fax: +385 1 6168 284
e-mail: mario.storga@fsb.hr
URL: www.cadlab.fsb.hr