

A Bayesian Approach to Robot Group Control

Tomislav Stipanovic, Bojan Jerbic and Petar Curkovic

Department of Robotics and Production System Automation, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Zagreb 10000, Croatia

Received: August 17, 2011 / Accepted: August 29, 2011 / Published: September 25, 2011.

Abstract: This paper describes a Bayesian approach to robot group control applied in industrial applications. The proposed model is based on well-known concepts of Ubiquitous Computing and can enable some degree of contextual perception of the environment. Compared with classical industrial robots, usually preprogrammed for a limited number of operations/actions, the system based on this model can react in uncertain situations and scenarios. The model combines ontology to describe the specific domain of interest and decision-making mechanisms based on Bayesian Networks to enable the work of a single robot without human intervention by learning Behavioral Patterns of other robots in the group. The described model is designed to be expressive enough to provide adequate level of abstractions needed for making timely appropriate actions and respecting the current application.

Key words: Robotics, Bayesian networks (BN), descriptive logic (DL), ontology, context, assembly.

1. Introduction

Robots in their essence have the purpose to replace human labor, not only in industrial applications but in other human activities, too. There are many applications connected to robotics, as for example in medicine, rescue operations, research, aiding the disabled people, etc. In order to fulfill the requirements of everyday life, robotic systems are inevitably becoming more and more sophisticated. The level of complexity demands novel or different research perspectives to be considered.

Ubiquitous Computing (UbiComp) [1] is a post-desktop model of human-computer or computer-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities [2]. The environment then becomes a space constantly analyzed by devices in

order to detect significant changes that can trigger the system to react, depending on its original functionality. UbiComp applications are normally envisioned to be sensitive to context, where context can include an object's location, activity, goals, resources, state of mind, and nearby people and things. Ubiquitous computing involves many different research areas, e.g., Distributed Computing, Mobile Computing [3], Sensor Networks, Human-Computer Interaction, Artificial Intelligence [4], etc. In an automatic assembly, the control of a system is usually connected to the control of the working environment. An uncontrolled situation is any situation where any object or subject is not completely defined from the aspects of position, orientation, action and/or process. Every environment is naturally unstructured, which can be revealed if it is observed under a fine enough scale. In other words, it is not possible to completely determine any environment, no matter how tight the applied tolerance ranges may be. Unconstrained environment is usually introduced to the system through the application of tolerances. This is connected with issues of sensitivity and instability and may result in malfunctioning, even if small

Bojan Jerbic, full professor, Ph.D., research fields: robotics, artificial intelligence, multi-agent systems.

Petar Curkovic, Ph.D., research fields: evolutionary algorithms, optimisation, multi-agent systems.

Corresponding author: Tomislav Stipanovic, B.Sc., head of lab, research fields: probabilistic robotics, knowledge representation, machine vision, multi-agent systems. E-mail: tomislav.stipanovic@fsb.hr.

environmental changes occur [5]. For such reasons, the system that cannot perceive the environment has to be programmed for a limited range of actions foreseen in advance by the system developer. By default, it cannot act in any unpredicted situation. In the best scenario, it could send a signal that an unpredicted situation had occurred. Dynamic information control, based on the contextual perception of the environment, needs less predetermined operational and structural knowledge. That requires system adaptation skills and some level of decision-making capabilities based on artificial intelligence methodologies. Each object, process or condition is unique by its very nature. Therefore, the context of space and time becomes an important task in autonomous system development. If an agent is able to make decisions about an action that is not completely restrained in the workspace, using certain perceptions, knowledge and intelligence, it can be said that the system is controlled [6]. In order to get an automatic system controlled, the corresponding knowledge about all relevant components and processes must exist. For this reason, the research of new methodologies and paradigms is directed toward the development of adaptive, anthropomatic and cognitive agent capabilities [7]. A model that relies on ontology for defining an industrial assembly/disassembly domain, Description Logic Reasoning (DL Reasoning) for planning an adaptive behavior and Bayesian Network Reasoning (BN Reasoning) for probabilistic action planning, can enable adaptive and autonomous behavior for all agents in the group. By utilizing the proposed model, a group of robots would become capable to convert their ordinary environment to a ubiquitous one. Behavioral patterns produced by the model can in their essence be compared to one that uses living beings for contextual understanding.

This paper is organized as follows: Section 2 introduces the Cognitive-Behavioral Model for Robot Group Control. This section represents the main core of the article. Sections 3-5 are used to describe the main model components: the environment, Core Ontology

and Bayesian Network Reasoning part of the model. Section 6 gives conclusions and possible future research directions.

2. Cognitive-Behavioral Model for Robot Group Control

The transition from free to controlled/known spatial state is one of the most demanding tasks in automatic processes. A programmer that tries to control the system has to cope with many certain and uncertain situations. Although it is possible, it is very hard to model a complex system to predict all possible outcomes that the environment is able to produce due to its nondeterministic nature. A system designed in such a manner can be called reactive because it reacts to environmental stimulus. The reactive system can be very fragile if something unexpected occurs, because it usually does not have self-recovery capabilities that would, by default, prevent errors arising from unexpected situations. On the contrary, the system that is able to realize a context of an environment can act depending on contextual information. Such a system can potentially do both: It can act reactively and it can comprehend the present and predict results of its future actions. The contextual perception implies understanding of a problem domain much broader than a single agent could feel and it can be carried out with the interaction between the agent and the environment together with related objects, other agents, processes and events.

Fig. 1 shows the proposed Cognitive-behavioral model for robot group control. As it can be seen in the figure, the model tends to be used in industrial environments for robot assembly/disassembly operations.

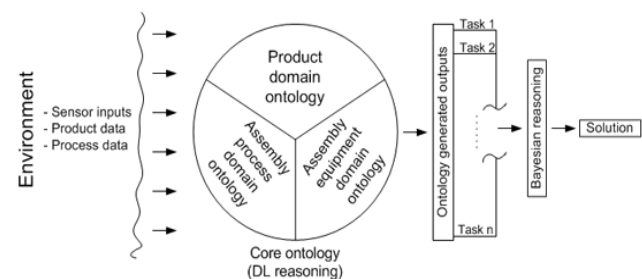


Fig. 1 The proposed model for robot group control.

The model includes the following components: information gathering by means of sensors integrated into the environment, domain ontology together with DL and BN Reasoning mechanisms used for finding the solution, depending on the Ontology Generated Outputs. The synergy of all model components can ensure the adaptive behavior of the robot working in the group with respect to current environmental conditions and predefined knowledge about a particular domain of interest. By predicting behavioral patterns of other robots in the group, a single robot becomes capable to plan its own behavior.

Another challenge is to develop a collaborative robot group work in real life scenarios. By using probabilities, it is possible to give an agent the capability to behave in seemingly non-predictable scenarios. First of all, the agent has to learn behavioral patterns of other agents by means of BN with respect to DL Reasoning and previously or simultaneously collected stimulus coming from the environment. After the learning phase, the agent can become capable of predicting actions coming from other agents and depending on its DL Reasoning mechanisms.

3. The Environment

How, when and which data to collect are important questions for all further steps leading to contextual perception of the environment. Answers depend mainly on application goals and require a thorough analysis to be made. The analysis should reveal spatial and temporal dependencies and characteristics of processes, equipment and objects. Information obtained by sensors is expected to change constantly. In robotic/automatic assembly, information about the position and orientation of work pieces and all other relevant in-process objects and processes represents essential data for program control. The change in information is a consequence of different environmental conditions, for example, change of position of the part to be assembled or change in information about the available part quantity. This can

be called expected informational change. The system has to decide only how to use information gathered autonomously with respect to used ontology. Fig. 2 shows the environment for model development and testing.

The used environment is characteristic for assembly/disassembly industrial applications and it contains different types of sensors for providing an information flow for decision-making processes with respect to context by using the ontology together with DL Reasoning.

4. Core Ontology

Ontologies denote a formal representation of entities (classes) along with associated attributes (objects) and their mutual relations [8]. Since ontologies allow representation of an arbitrary domain and can simplify work for end users, they have been proven to be extremely convenient. With time, ontologies have also been proven to have certain disadvantages. By designing a certain domain coming from his own field of expertise, an expert uses personal knowledge and impressions, which can differ when compared with other experts in the field. Such an approach can disable



Fig. 2 The environment used for model developing and testing. It contains six axes robots, assembly line with stopping places, vision systems, optical, laser and capacitive sensors, parts to be assembled along with pallets and product carriers.

other ontologies representing the same domain to exchange information mutually. This means that gaps, overlaps, and inconsistencies will continue to exist when independently developed ontologies are used together [9].

Projects MASON [10] and ONTOMAS [11] represent an endeavor of researchers to build ontology for describing the field of production activities. The design process of the Core Ontology was guided by a concern about integration with other similar ontologies. We can draw such a conclusion from the fact that the ontology developed through ONTOMAS project and the Core Ontology uses the same Integrated Assembly Model for domain knowledge description, originally proposed by Rampersad [12]. Fig. 3 shows a part of the Core Ontology and represents the taxonomy of Assembly Operations.

The next step after the taxonomy definition is to transform it to ontology by using DL. The developed ontology can then be integrated into the Core Ontology along with all other domain definitions.

The model is designed in a way that every assembly operation represents an algorithm for controlling the robot. For example, the “pick up” operation is a part of “handling” the group and contains predefined actions for picking the part previously seen by means of the vision system. The algorithm for simple “pick and place” operations, which is written to be executed on FANUC robots (FANUC Robotics—<http://www.fanurobotics.com/>), is shown below.

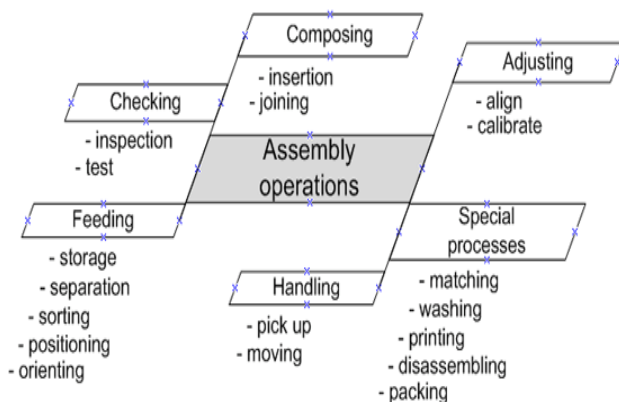


Fig. 3 Assembly operations taxonomy represents the part of Core Ontology.

```
//operation PICK_UP
program PICK_UP
1.UFRAME_NUM = 8
2.U TOOL_NUM = 2
3.R [1] = DI [1]
4.CALL INSPECTION
5.PR[R [2]] 100mm/sec FINE
6.R [10] = OFF
7.R [11] = ON
8.WAIT 0.1
9.R [10] = ON
10.R [11] = OFF
[End]

// operation INSPECTION called from the PICK_UP
operation program INSPECTION
1.VISION RUN_FIND 'OBSERVER'
2.VISION GET_OFFSET 'OBSERVER' VR [1]
3.R [2] = VR [1].MES[1]
[End]
```

Every elementary operation defined in the Core Ontology has its own algorithm written as a program structure understandable to robots. An agent, or let us say a robot, would start the programs depending on the current contextual information derived from interactions between all domain components with respect to the Core Ontology. One of the most important features is the model’s ability to support decision-making processes. DL Reasoning would assure a correct order of the program executions for reactive robot group behavior (Fig. 4).

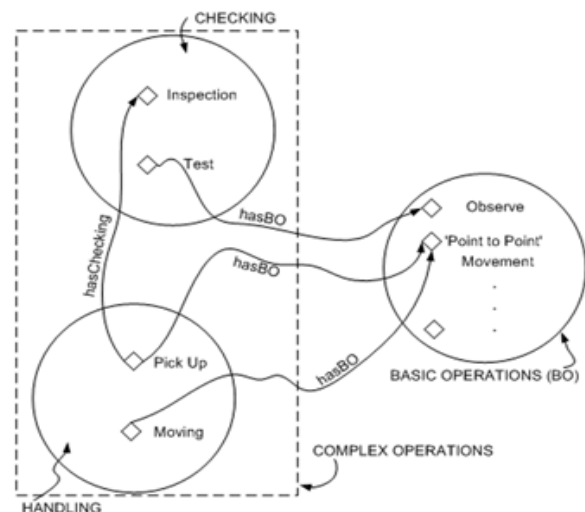


Fig. 4 A part of the assembly process domain ontology showing mutual relationships between classes, objects and properties.

Fig. 4 shows the way how previously defined taxonomy can be transformed into ontology for defining the working environment which is characteristic to robotic assembly operations. By containing all the properties, objects, and classes, a well-defined ontology can represent the knowledge that can be used for system suggesting the best possible solutions with respect to current environmental conditions and needs.

The Core Ontology part of the model along with DL Reasoning is expressed using OWL-DL [13]. OWL-DL is one of the OWL [14] dialects that support knowledge sharing and reuse, which can be very important if we want to add new knowledge to the model.

Fig. 5 shows the interpretation designed to define a part of the working environment characteristic for industrial assembly applications. Such interpretations have to contain all relevant components along with their corresponding properties and should be understandable to all system components. As can be seen from the figure, the interpretation is presented with a set of mutually connected nodes. Typically, nodes are used to characterize concepts, i.e., sets or classes of individual objects, and links are used to characterize relationships among them [15]. So called atomic concepts are represented by literals *a*, *b*, ..., *f* and include other classes and individuals. For example, object PRODUCT B is a member of three other classes (*c*, COMPLICATED and ITEM). DL enables to produce

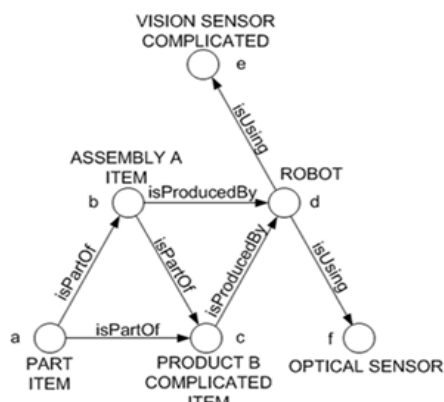


Fig. 5 A partial interpretation of the working environment.

other classes (concepts) derived from interpretations by means of DL descriptions. Such complex descriptions can contain other concepts (classes), roles (properties) and constructors. The same DL methodologies are used within Protégé-OWL editor [16-17] to derive proper system responses according to the information gathered from the working environment.

5. Bayesian Network Reasoning

Probabilities techniques can help the system work in uncertain situations or scenarios. With respect to the proposed model, DL Reasoning can provide more than one possible solution, which can enable indecisive behavior of robots. In a non-deterministic world, the deterministic way of seeing the world is often not expressive enough to address real world problems [18]. Mathematically, a Bayesian Network is a directed acyclic graph in which a set of random variables makes up the nodes in the network. A set of directed links connects pairs of nodes, and each node has a Conditional Probability Table that quantifies the effects of parents on it.

The ontology can produce ambiguities in its solutions by suggesting more than one solution for solving the task. BN can be used in determining the best option with respect to current stimulus coming from sensors placed in the environment. Each of the variables in the BN is represented by nodes. A single node in the network contains a set of probable values called states for each variable. Fig. 5 shows one BN along with corresponding stages for Behavioral Patterns and three robots.

The Behavioral Pattern (BP) represents a sequence of elementary operations derived from DL Reasoning, e.g., ABCD sequence can denote: A—inspection, B—pick up, C—insertion and D—testing. By gathering information about BP together with other relevant spatial and temporal information coming from sensors, we can calculate the so called posterior probability $p(x|y)$. If we want to infer a quantity x by means of sensory data y , we can use Bayes' rule to

determine the inverse probability, which specifies the probability of data y assuming x was the case, as in (1):

$$p(x|y) = \frac{p(y|x) \times p(x)}{p(y)} \quad (1)$$

In robotics, this inverse probability is often coined “generative model,” since it describes, at some level of abstraction, how state variables X cause sensor measurements Y [19].

Terms a priori and a posteriori are usually used in philosophy to distinguish two types of knowledge, justification or arguments. A priori knowledge or justification is independent of experience as opposed to a posteriori knowledge which makes reference to experience and can be used for decision making processes [20].

By the given environmental conditions, the ontology can suggest a set of different behavioral patterns. The second problem is very interesting and it can be seen in mutual interactions of the robots belonging to the group. Collisions are more than certain because all robots use the same ontology for planning reactive behaviors. For example, if the ontology offers the group the behavioral pattern ACDG, which indicates a sequence of operations for solving the current task, the question is how to decide which robot should carry out the task. The Cognitive-Behavioral Model for Robot Group Control uses BN to solve the riddle. By anticipating the next steps of other robots in the group, a single robot can plan its future actions. Before they are ready, robots in the group should take the learning stage. At the beginning, the group should contain only one member to react in the manner derived from the ontology. Before joining the group, the second robot should learn how to predict behavioral patterns of the first robot by observing it. If ontology proposes a couple of behavioral patterns for a certain action scenario, the second robot should determine its behavioral pattern respecting the ontology solutions and the pattern used by the first robot. The process of integration is the same for every new robot joining the group with a small difference. The new robot should

observe both, the behavioral pattern used by the first and by the second robot (Fig. 6).

The table of influence (Table 1) shows the mutual influences between all system components.

The probability of any node belonging to BN is described using Condition Probability Table (CPT). Probabilities on a particular node are affected by the state of other nodes depending on prior information about the relationships. Fig. 7 shows the CPT table together with related conditional probabilities between Robot 01 and BP.

Information about conditional probabilities for each robot has to be defined in advance. By altering this information, the system designer gets the opportunity

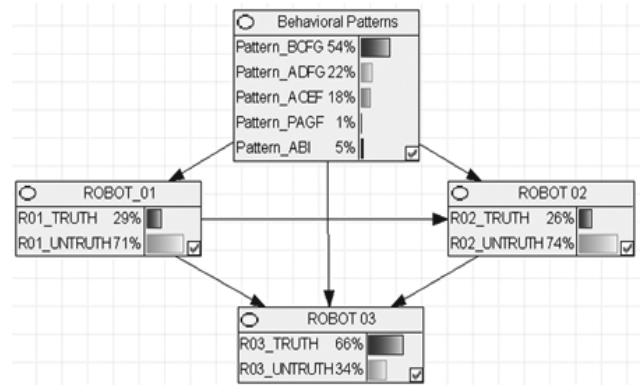


Fig. 6 BN is used as a mechanism to plan future behaviors for three robots.

Table 1 The table of influence can reveal the hierarchy for the member of a domain. Ontology generated responses are at the first place (Behavioral Patterns—BP) and all other system components are influenced by them. Robot 01 is influenced only by BP. Robot 02 is influenced by Robot 01 and BP. Robot 03 is influenced by all of them.

	BP	Robot 01	Robot 02	Robot 03
Robot 01	+			
Robot 02	+	+		
Robot 03	+	+	+	

The screenshot shows the 'Node properties: ROBOT_01' window with a CPT table. The table has columns for Behavioral Patterns (Pattern_BCFG, Pattern_ADFG, Pattern_ACEF, Pattern_PAGF, Pattern_ABI) and rows for R01_TRUTH and R01_UNTRUTH.

Behavioral Patterns	Pattern_BCFG	Pattern_ADFG	Pattern_ACEF	Pattern_PAGF	Pattern_ABI
R01_TRUTH	0.11	0.64	0.42	0.81	0.1
R01_UNTRUTH	0.89	0.36	0.58	0.19	0.9

Fig. 7 CPT table indicates how certain BP influences the robots.

to define system priorities and to achieve certain goals of assembly processes. By using this knowledge, each robot should be able to predict its next actions. The last robot joining the group has the most information about conditional probabilities, because it depends on choices of all other robots.

The next algorithm is used to describe a procedure that a single robot has to take while determining its further actions respecting the proposed and previously utilized Behavioral Patterns.

```
//Working procedure for
//a single robot joining the group.
Capture ontology generated outputs
Call Bayesian Reasoning Mechanism
Check for pattern number
if (pattern_number > 0) then
  do while not goal
  Use the pattern with highest probability
endif
if (pattern_detected == true and pattern_number > 0)
  then
  do while not goal
  Set robot to execute the pattern
  Update Bayesian Network
endif
```

6. Conclusions

In this paper a robot group control model based on Bayesian Reasoning is discussed. Cognition of the environment is achieved by means of ontology, which is suitable for knowledge storing, sharing and reuse. In order to give the group a behavioral component, a probabilistic approach based on Bayesian Network is used. That enables a single robot to plan its behavior according to other parameters: other robots, which belong to the group and all other relevant environmental components. By observing the proposed model, a couple of conclusions should be made. Deterministic chaos inevitably obstructs absolute expectations, always producing slightly changed situations. To alter uncertain situation, conventional

automation methods tend to create technical systems as almost perfect constructions. It seems that such efforts are definitely hopeless and result in expensive and inefficient systems. Such an approach raises more issues that affect almost every contemporary industrial factory in the world: A lack of space and rigidity of the production systems. These problems are even more prominent in Europe. Making industrial systems adaptive, small, cheap and competitive with the rest of the world is a question that appeared many years ago. Indeed, the system based on contextual perception of the environment can be converted to work on other similar tasks relatively easily compared to classical industrial production lines. A robot with such properties must be able to interpret and understand the context of the environment in order to adapt its strategies to effectively work in a group. The Cognitive-Behavioral Model for Robot Group Control enables some degree of contextual understanding and provides a way for the robot to plan its actions by observing other robots. Probabilistic reasoning based on BN can be used to increase the level of security by learning and anticipating behavioral patterns of all other robots. Such an approach could also increase the overall security level of the system.

There are a few more things that can be analyzed and implemented into the original model. It is possible to identify certain uncertainties in other parts of the Core Ontology. For example, by altering the Assembly Process Domain Ontology it is possible to gain certain production goals by trying to optimize particular production parameters.

Deterministic chaos should be accepted as a natural phenomenon and the development philosophy changes toward the development of intelligent machines capable of adapting their behavior according to the natural imperfect world where nothing is absolutely ideal or accurate.

References

- [1] Wikipedia, available online at: http://en.wikipedia.org/wiki/Ubiquitous_computing, March 2011.

- [2] Mark Weiser's Home Page, available online at: <http://www.ubiq.com/hypertext/weiser/UbiHome.html>, March 2011.
- [3] P. Curkovic, B. Jerbic, T. Stipanovic, Swarm-based approach to path planning using honey-bees mating algorithm and ART neural network, *Solid State Phenomena* 147-149 (2009) 74-79.
- [4] P. Curkovic, B. Jerbic, T. Stipanovic, Hybridization of adaptive genetic algorithm and ART 1 neural architecture for efficient path planning of a mobile robot, *Transactions of FAMENA* 32 (2) (2008) 11-21.
- [5] T. Stipanovic, B. Jerbic, Self-adaptable vision system, in: *DoCEIS'10—Doctoral Conference on Computing, Electrical and Industrial Systems*, Lisbon, Portugal, 2010.
- [6] T. Stipanovic, P. Curkovic, B. Jerbić, Robust autonomous assembly in environment with relatively high level of uncertainty, in: *MITIP 2008*, S.R.O. Press, Czech Republic Bily Slon, 2008, pp. 193-198.
- [7] M. Svaco, B. Sekoranja, B. Jerbić, Autonomous planning framework for distributed multiagent robotic systems, *Technological Innovation for Sustainability*, Heidelberg: Springer Verlag, 2011, pp. 147-154.
- [8] T. Gruber, Toward Principles for the design of ontologies used for knowledge sharing, *International Journal Human-Computer Studies* 43 (1994) 907-928.
- [9] P.C.G. Costa, K.B. Laskey, G. Alghamdi, Bayesian ontologies in AI Systems, in: *Proceedings of the Fourth Bayesian Modelling Applications Workshop*, held at the Twenty Second Conference on Uncertainty in Artificial Intelligence, Cambridge, MA, USA, 2006.
- [10] S. Lemaignan, A. Siadat, J.Y. Dantan, A. Semenenko, MASON: a proposal for an ontology of manufacturing domain, *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 2006, pp. 195-200.
- [11] N. Lohse, Towards an ontology framework for the integrated design of modular assembly systems, Ph.D. Thesis, University of Nottingham, 2006.
- [12] H.K. Rampersad, *Integrated and Simultaneous Design for Robotic Assembly*, John Wiley & Sons Ltd., Chichester, ISBN 0-471-95018-1, 1994.
- [13] Y. Al-Safi, V. Vyatkin, An ontology-based reconfiguration agent for intelligent mechatronic systems, in: V. Marik, V. Vyatkin, A. Colombo (Eds.), *Holonic and Multi-Agent Systems for Manufacturing*, Lecture Notes in Computer Science, Vol. 4659, 2007, pp. 114-126.
- [14] World Wide Web Consortium (W3C), available online at: <http://www.w3.org/standards/techs/owl>, accessed: March, 2011.
- [15] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, Applications*, Cambridge University Press, Cambridge, UK, 2003, p. 6.
- [16] H. Knublauch, R.W. Fergerson, N.F. Noy, M.A. Musen, The protégé OWL plugin: an open development environment for semantic web applications, in: *International Semantic Web Conference 2004*, Hiroshima, Japan, 2004, pp. 229-243.
- [17] The Protégé Ontology Editor and Knowledge Acquisition System Home Page, available online at: <http://protege.stanford.edu/>, accessed: April 2011.
- [18] A. Larik, S. Haider, Efforts to blend ontology with bayesian networks: an overview, in: *Proceedings of 3rd International Conference on Advanced Computer Theory and Engineering*, Chengdu, China, 2010.
- [19] S. Thurn, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, 2006.
- [20] Wikipedia, available online at: http://en.wikipedia.org/wiki/A_priori_and_a_posteriori, accessed: March 2011.