8-6-2011

A Meta-Model Ontology based on Scenarios

Andrey Soares
Southern Illinois University Carbondale, asoares@siu.edu

Frederico Fonseca
The Pennsylvania State University, fredfonseca@ist.psu.edu

A Meta-Model Ontology based on Scenarios

Andrey Soares
Southern Illinois University Carbondale
asoares@siu.edu

Frederico Fonseca
Pennsylvania State University
fredfonseca@ist.psu.edu

ABSTRACT

This paper describes a proof-of-concept designed to test a meta-model ontology based on scenarios for representing the procedural knowledge of a given domain. The research was conducted as part of a study on methodologies to build ontologies for Information Systems. The frequent use of scenarios in the initial steps of the methodologies motivated us to investigate its use in the process of building ontologies for IS modeling. This research proposes the use of the components of scenarios as constructs of a meta-model ontology. With this approach, the process of building scenarios about a domain is mingled with the process of building domain ontologies.

Keywords

Meta-Model, Ontology, Scenario, Procedural Knowledge, Information Systems

INTRODUCTION

This research reports a scenario-based approach to augment the modeling of procedural knowledge in methodologies to build ontologies for Information Systems (IS). The research proposes the use of a meta-model ontology based on scenarios to represent procedural knowledge and temporal relations of a domain.

Scenarios have extensively been used to acquire the knowledge used to build domain ontologies (Grüninger et al. 1995; Lee 2006). However, they are usually discarded once the harvest of concepts and relationships is completed. We propose a scenario-based approach that mingles the process of building scenarios about a domain with the process of building domain ontologies. In particular, we suggest using the components of scenarios as ontological constructs of a meta-model ontology. We argue that a domain ontology created with this approach is enhanced by including knowledge about not only what the domain is but also how the domain works.

BACKGROUND

The research was conducted as part of a study on methodologies to build ontologies for Information Systems. The study targeted the area of Ontology-Driven Information Systems (ODIS), where ontology plays a central role both at development time and at run time of Information Systems (Guarino 1998). The term methodology used in the study follows the terminological relationships in methodologies presented by Gómez-Pérez et al. (2004, p.109). They consider that a methodology is composed of methods and techniques.

The study was motivated by (1) the increasing use of ontologies in Information Systems and the proliferation of conceptual modeling methods lacking theoretical foundations (Wand et al. 1989), (2) the fact that after three decades of research and a shared understanding that ontology plays a central role in Information Systems, researchers have not yet produced comprehensive guidelines for building ontologies for Information Systems Analysis and Design (ISAD), (3) Cardoso's survey (Cardoso 2007), which shows that 60% of the participants did not use any methodology to develop ontologies, (4) the fact that the process of building ontologies is considered an art rather than an engineering process (Gómez-Pérez et al. 2004), and (5) the results of our preliminary research on building ontology, which revealed four issues to be considered when building ontologies for IS. The issues identified are related to the need of approaches to support:

- Meta-models: a high-level structure that provides a framed view of the domain, guides the construction of domain ontologies, and increases the semantic for understanding a domain.
- Procedural knowledge: describes a set of tasks that need to be performed for achieving a goal.
- Temporal relations: represent the chronological arrangement of the tasks and their dependencies.
- Knowledge acquisition: relates to a systematic approach for capturing the domain knowledge to be represented by the ontology.

Based on the concerns above, we set up a research to investigate existing methodologies that could provide methodological approaches to overcome the issues raised. The research adopted a formal method, called Systematic Reviews (Kitchenham 2004), to conduct a literature review of methodologies to build ontologies. We searched publications from major

bibliographic databases (i.e., ACM, IEEE, Springer, Elsevier, Web of Science, and Proquest) from which we selected 30 methodologies to investigate (see Figure 1) as result of our search strategy. The timeline refers to the year of the publication presenting the methodology, rather than the year when the methodology was created. The methodologies were analyzed with regard to the twelve categories that we developed. The categories cover mainly the core components of an ontology (i.e., concepts, properties, relationships and axioms), and the issues identified from our preliminary research. The results of the systematic review are beyond the scope of this paper, and will be discussed elsewhere.

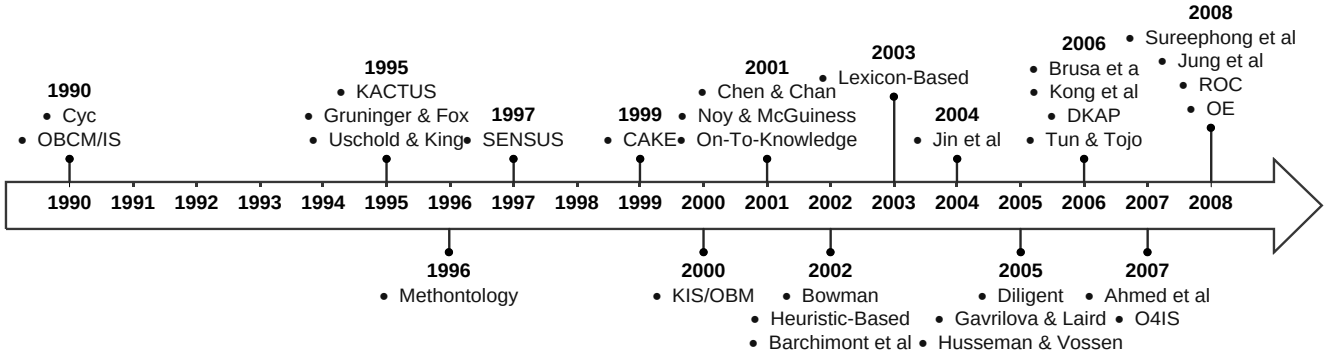

Figure 1: Development timeline of the selected methodologies

Procedure Knowledge

Wand & Weber (1989) state "that a modeling scheme must represent two aspects of a system: structure (statics) and behavior (dynamics)" (p.83). The knowledge involved with the static and dynamic aspects of a system are often referred to as conceptual knowledge (i.e., concepts and relations) and procedural knowledge (i.e., processes and tasks) (Milton 2007).

The procedural knowledge is usually neglected as a domain knowledge to be included in an ontology for Information Systems. In fact, 19 out of the 30 methodologies investigated did not provide support to identify and represent this kind of knowledge. This is a kind of knowledge of "processes, tasks and activities", especially of how to perform tasks and related sub-tasks (Milton 2007, p.4).

Some of the methodologies investigated provided support to procedure knowledge. The Cyc methodology (Lenat et al. 1990), for example, describes constructs (e.g., event, process, state, etc.). The Gruninger & Fox methodology suggests the use of competency questions related to planning and scheduling, which refers to "what sequence of activities must be completed to achieve some goal?" (Grüninger et al. 1995, p.4). The O4IS methodology (Vandana 2007) proposes to extract procedural knowledge from scenarios (or use case scenarios). However, scenarios are not directly represented within the ontology. Instead, the knowledge of scenarios is used as input to other artifacts.

Other methodologies presented methods to identify procedural knowledge. For instance, the O4IS methodology proposes a Procedural Concept View to describe "the knowledge (procedural knowledge) about the emotional states, intentions, plans and rules" of the domain (Vandana 2007, p.118), and the CAKE methodology (Gavrilova et al. 1999) proposes the use of a stratification process, which includes the HOWTO-Knowledge (i.e., functional analysis) and the WHY-knowledge (i.e., causal analysis). The O4IS methodology presents the most comprehensive approach for the analysis and representation of procedural knowledge. The methodology presents the SAR:Functional Relationship, which is based on the Verb-Phrase Ontology (Storey et al. 2004) to identify actions and its dependences. In addition, O4IS uses the ECA (i.e., Event-Condition-Action) Rule combined with the 5Ws (i.e., who, what, where, when, why) to identify and to represent procedural knowledge.

Temporal Relations

A temporal relation is an approach in knowledge representation involving the events of an application domain, as "they exhibit a history of changes through time" (Mylopoulos et al. 1990, p.7). These changes in the application domain can be described in terms of how events relate to each other. In this case, temporal relations are used to provide the ordering of the events (i.e., temporal constraints) and to document the changes occurring in a domain. Despite this integration, and the fact that 11 methodologies have provided some support to the process of identifying tasks, only five out of the 30 methodologies provided support to the identification and representation of temporal relations between tasks. Perhaps, as Guarino (1998) explains, "some parts of this [domain] knowledge are encoded in the static part of the program in the form of type or class declarations" (p.13), and are not explicitly represented in the ontology.

The O4IS methodology bases their SAR:Temporal Relationships approach on the linear temporal logic theory, which describes the relation between two events (e.g., event A starts before event B). It uses the primitives follow/precede and requires to represent dependencies between two actions. The Gruninger & Fox methodology identifies temporality through a set of informal competency questions called Temporal Projection, which is one of the set of questions to create the Activity Ontology (i.e., actions performed). For instance, "given a set of actions that occur at different points in the future, what are the properties of resources and activities at arbitrary points in time?" (Grüninger et al. 1995, p.3). The Cake methodology suggests the stratification WHEN-knowledge for Temporal Analysis of the domain. That would include "Schedules, Time Constraints, etc." (Gavrilova et al. 1999, p.753).

Temporal constraint is also proposed by the Heuristic-based methodology (Sugumaran et al. 2002) as part of the heuristic-3.2. This heuristic states that "one term/relationship must occur before another" (Sugumaran et al. 2002, p.259). The methodology also includes heuristics for mutual inclusive constraint (i.e., heuristic-3.3), where "one term/relationship requires another for its existence" (p.259); and for mutual exclusive constraint (i.e., heuristic-3.4), where "one term/relationship cannot occur at the same time as another" (p.260). The use of scenarios, as proposed in this paper, could provide information about temporal relations within a domain through the sequence of tasks of a scenario and their dependences.

Scenarios

Scenarios have been successfully used in IS design for identifying and capturing domain knowledge (Weidenhaupt et al. 1998), and have already been adopted for ontology design (Grüninger et al. 1995; Lee 2006). According to Grüninger & Fox (1995), "any proposal for a new ontology or extension to an ontology must describe the motivating scenario, and the set of intended solutions to the problems presented in the scenario. This is essential to provide rationale for the objects [and their relations] in an ontology" (p.2). They also consider that through scenarios "we can understand the motivation for the proposed ontology in terms of its applications" (p.2).

The frequent use of scenarios in the initial steps of the methodologies motivated us to further investigate its use in the process of building ontologies for IS modeling. In this research we promote the use of scenarios not only as mechanisms for eliciting knowledge for building ontologies, but also as ontological constructs. A thorough investigation of the components of a scenario revealed a prospective structure for using it as a meta-model ontology. Thus, instead of taking the traditional approach of textual narratives, the research explored the use of scenarios in the form of formal ontologies, where the components of a scenario become the constructs of a meta-model ontology. Textual descriptions are explicit source of knowledge for humans (Uschold 2003), nevertheless we also want this knowledge to be used by machines. We argue that having a meta-model ontology based on scenarios could provide a richer semantic for machines to understand content meaning. Without this layer of understanding on top of the domain ontology (i.e., ontological view of a domain), meaning would be hardwired into the application software (Uschold 2003).

Related Work

The use of a Scenario Ontology to frame knowledge acquisition has already been proposed by Yu-N & Abidi (2000); however, their approach does not cover the creation of domain ontologies. Instead, it creates instances of scenarios, which are represented with XML. According to Yu-N & Abidi, the scenario instances could be translated, with the help of the scenario ontology, into other representation languages, such as Prolog. Another use of scenario ontology includes selecting concepts from scenarios written as textual narratives, and adding the concepts into an ontology called Scenario Ontology (Polhill et al. 2006). In this research, we propose the integration of the processes of building scenarios and ontologies.

PROOF-OF-CONCEPT

Figure 2 illustrates the envisioned meta-model ontology (i.e., Scenario Ontology), connecting with a domain ontology. The meta-model ontology accounts for the description of scenarios, their goals and specific events, which would support a description of the procedural knowledge embedded in a domain. The dashed lines represent the mapping between the levels, which can be achieved by adding the domain concepts under the structure of the meta-model concepts. This way, we can distinguish, for example, the class Adopter (i.e., a type of Agent) and the class Cage (i.e., a type of Resource).

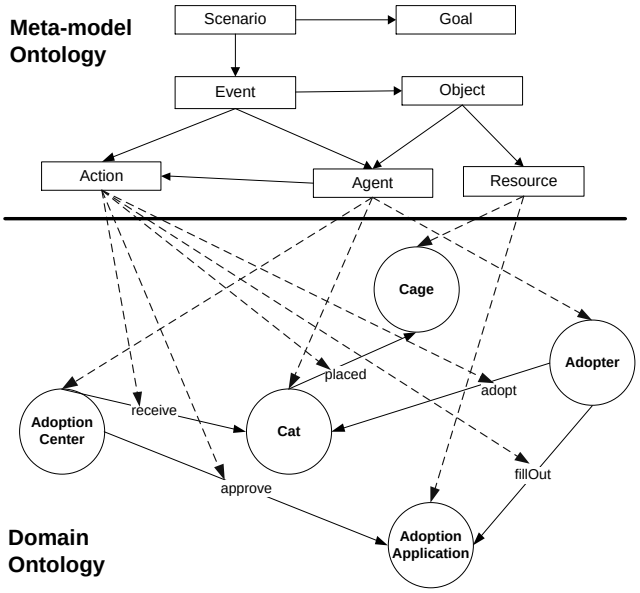

Figure 2: Envisioned mapping between the meta-model ontology and the domain ontology

The proof-of-concept was composed of the following main steps: (1) build the meta-model Ontology (i.e. Scenario Ontology), (2) create a domain ontology and import the Scenario Ontology into it, and (3) represent a sample scenario in the domain of an animal shelter by mapping the domain concepts into concepts of the Scenario Ontology. The proof-of-concept was implemented with the Protégé Ontology Editor v.3.3.1 (http://protege.stanford.edu), a popular editor to handle the language used to represent ontology, OWL-Web Ontology Language (http://www.w3.org/2004/OWL).

The scenario structure used in this research (see Figure 3) is based on the structure presented by Rolland et al. (1998b, p.8). However, we added a property called "predecessor" to help representing the temporality of events, we added a component "action" to distinguish the actions within an event, and we did not follow their separation between normal and exceptional scenarios. A scenario has a goal that is achieved by performing one or more events. A scenario has pre-conditions that initiate the scenario, and post-conditions that are reached upon completion of a scenario. An event can be an atomic event (i.e., a single event) or a flow of events (i.e., an event composed of sub-events). An event can be dependent on the occurrence of other events (i.e., predecessor). The description of an event includes an agent (i.e., subject) that performs an action (i.e., verb) upon an object (i.e., object direct and object indirect), which can be another agent or a resource.

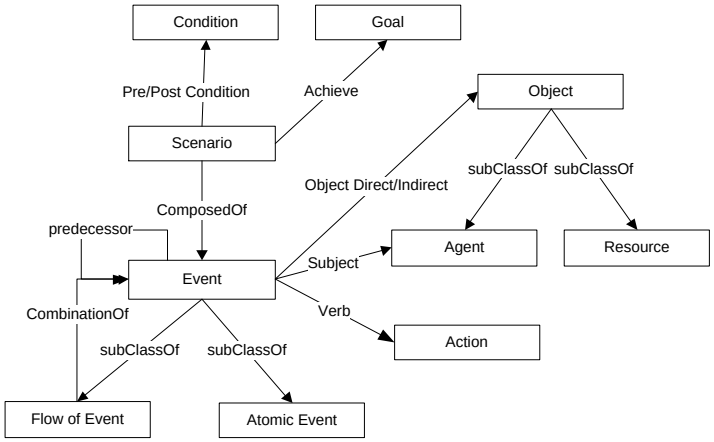

Figure 3: Scenario-based Meta-model ontology

With the meta-model ontology in place, we were able to represent concepts and relationships regarding the animal shelter domain. The Scenario Ontology becomes a meta-model ontology, which reflects a commitment to view the domain as scenarios. To help identify concepts and scenarios, we used Task Analysis to decompose complex events into sub-events

(Rosson et al. 2001). The resulting clusters of tasks turned into our scenarios, and the tasks turned into our events. Constraints between tasks, such as repetition, have not been addressed in the proof-of-concept.

Following the example of linguistic clause patterns (Rolland et al. 1998a), we expressed an event with subject = Volunteer, verb = place, objectDirect = Cat, and objectIndirect = Cage. An instance of an event labeled "Volunteer_place_Cat_in_Cage" is presented in Figure 4. From this instance, humans should be able to understand roughly that a volunteer named Joseph placed a cat named Connor in a cage identified by the number 2. Machines, however, would need more details about the meaning of the action "place". The properties subject, objectDirect, and objectIndirect are restricted with *allValuesFrom* to enforce that only instances of the concepts defined can be part of the event. The concept "place" is restricted with *hasValue*, which means that a specific value is being assigned to the property verb. So, the restriction reads, only one action is permitted in this event and the action must be "place".

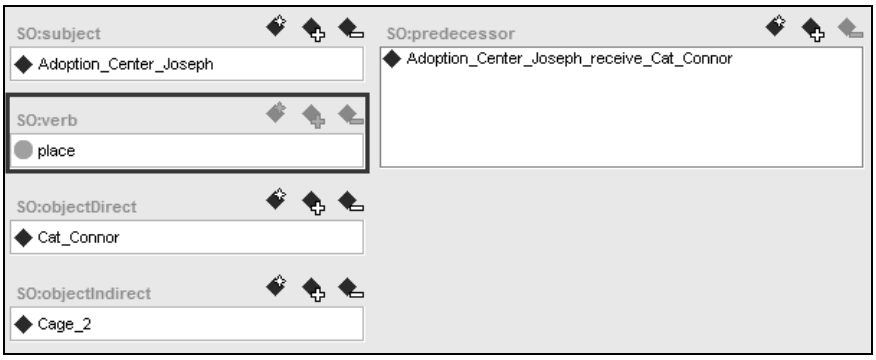

Figure 4: Instance of an atomic event for when a volunteer places a cat in a cage

Inspired by how a Gantt Chart (Wilson 2003) connects its tasks, we included a property called "predecessor" in the event. The property in each event provides information about the ordering in which the events are expected to be performed, and how the events depend on each other. In some cases, events within a scenario may be dependent on the completion of events from other scenarios. For instance, the event of a cat coordinator sending out the weekly cat census should trigger both the scenario for updating the website with new cats for adoption, and the scenario for taking pictures of the new cats. However, at some point, completing the website update will depend on receiving pictures of the cat. The property predecessor is also restricted with *allValuesFrom* to enforce that an instance of this event occurs after an instance of another event has occurred.

After creating the events, we can represent a sample scenario about a new cat arriving at the shelter. The scenario is triggered when someone brings, for example, a homeless cat to the animal shelter. A volunteer (e.g., adoption center representative) receives the cat and places the animal in a cage. Then, the volunteer registers some information about the cat, and makes a tag that will be displayed on the cage. Figure 5 shows the properties and restrictions used to define this scenario.

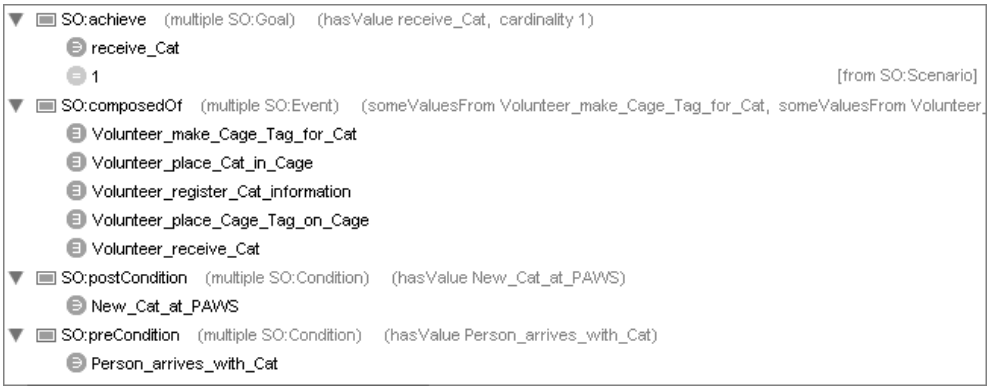

Figure 5: Definition of a scenario for when a new cat arrives at the animal shelter

CONCLUSION

This paper describes a scenario-based approach to enhance the process of building ontologies for Information Systems. The initiative resulted from our analysis of existing methodologies, where we observed a frequent use of scenarios in the initial steps of some methodologies. Thus, we were interested in finding out how scenarios could be used to improve existing

approaches. Our suggestion is to use the components of scenarios as ontological constructs to enhance the domain ontology by including a representation of procedural knowledge and temporal relations.

The use of a meta-model ontology based on scenarios is a valuable mechanism to build domain ontologies suitable to IS modeling. In particular, this approach could (1) enhance the process of acquiring and representing the procedural knowledge of a given domain, (2) provide a frame to view the domain as a system with goals and events, (3) help domain experts and ontology designers to identify domain concepts to be represented, and (4) augment the steps of some of the methodologies investigated, especially because the use of scenarios as promoted here addresses all four issues raised in the study.

With this Scenario-based approach, the process of representing scenarios is already part of the process of creating domain ontologies, which means fewer steps to build an ontology and no need to translate, for example, scenarios into ontologies. This approach should be for the most part a seamless transition with regard to what type of information to represent, yet it should be flexible to allow representation of individual types of information. For example, we could start by creating a list of terms and then using the terms to create scenarios. The formalized representation with ontologies should allow machines to handle the content of scenario, especially with regard to knowledge reuse, and automated views of the systems. For instance, an application handling the Scenario-based approach should be able to produce, with the knowledge from the ontologies, a view of a specific scenario with its tasks and agents, or a view of all scenarios that include a specific agent.

We share the view of an ontology being transformed into IS components (Kishore et al. 2004; Uschold 2008) as the ontology contains information about how the system works, and what information is needed and when. That is, the ontology should be able to provide information about the application programs, the interfaces and the information resources of an Information Systems. We see the representation of procedure knowledge and temporal relations as a considerable part for the success of ODIS. Uschold (2008) envisions ODIS with models that "will not go through the usual steps of code generation, compiling and linking. Rather these models will be already executing as they are being built" (p.14). For that to be achieved, we argue that ontologies should carry more information related to the control and execution of tasks within a domain.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for the constructive suggestions. This paper is based on a Ph.D. dissertation, completed in December 2009 at the Pennsylvania State University.

REFERENCES

1. Cardoso, J. "The Semantic Web Vision: Where are We?," *IEEE Intelligent Systems* (22:5), September/October 2007, pp 84-88.
2. Carroll, J.M. (ed.) *Scenario-based Design: Envision Work and Technology in System Development*. John Wiley and Sons, New York, NY, 1995.
3. Gavrilova, T., Voinov, A., and Vasilyeva, E. "Visual knowledge engineering as a cognitive tool " International Work-Conference on Artificial and Natural Neural Networks, IWANN'99, Springer Berlin / Heidelberg, Alicante, Spain, 1999, pp. 750-758.
4. Gómez-Pérez, A., Fernandez-Lopez, M., and Corcho, O. *Ontological Engineering: with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*, (1st ed.) Springer Berlin Heidelberg, 2004.
5. Grüninger, M., and Fox, M.S. "Methodology for the design and evaluation of ontologies," IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995, pp. 6.1–6.10.
6. Guarino, N. "Formal Ontology in Information Systems," Formal Ontology in Information Systems (FOIS'98), Amsterdam, IOS Press, Trento, Italy, 1998, pp. 3-15.
7. Kishore, R., Sharman, R., and Ramesh, R. "Computational Ontologies and Information Systems: I. Foundations," *Communications of the Association for Information Systems* (14:8) 2004, pp 158-183.
8. Kitchenham, B. "Procedures for Performing Systematic Reviews," Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1.
9. Lee, J. "The roles of scenario use in ontology development.," *Knowledge and Process Management* (13:4) 2006, pp 270-284.
10. Lenat, D.B., and Guha, R.V. *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*, (1st ed.) Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1990.
11. Milton, N.R. *Knowledge Acquisition in Practice: A Step-by-Step Guide* Springer-Verlag London, 2007.
12. Mylopoulos, J., Borgida, A., Jarke, M., and Koubarakis, M. "Telos: Representing Knowledge About Information Systems," *Information Systems* (8:4) 1990, pp 325-362.

13. Polhill, J.G., and Ziervogel, G. "Using ontologies with case studies: an end-user perspective on OWL," in: *Second International Conference on e-Social Science (NceSS 2006)*, Manchester, UK, 2006.
14. Rolland, C., and Achour, C.B. "Guiding the Construction of Textual Use Case Specifications," *Data & Knowledge Engineering Journal* (25:1-2) 1998a, pp 125-160.
15. Rolland, C., Souveyet, C., and Achour, C.B. "Guidind goal modeling sing scenarios," *IEEE Transaction on Software Engineering* (24:12) 1998b, pp 1055-1071.
16. Rosson, M.B., and Carroll, J.M. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*, (1st ed.) Morgan Kaufmann, Redwood City, CA, 2001, p. 448.
17. Storey, V., and Purao, S. "Understanding relationships: Classifying verb phrase semantics," in: *Conceptual Modeling – ER 2004. Proceedings of the 23rd International Conference on Conceptual Modeling*, P. Atzeni, W. Chu, H. Lu, S. Zhou and T. Ling (eds.), Springer-Verlag Berlin Heidelberg 2004, pp. 336-347.
18. Sugumaran, V., and Storey, V.C. "Ontologies for conceptual modeling: their creation, use, and management," *Data & Knowledge Engineering* (42:3) 2002, pp 251-271.
19. Uschold, M. "Where are the Semantics in the Semantic Web?," *AI Magazine* (24:3) 2003, pp 25-36.
20. Uschold, M. "Ontology-Driven Information Systems: Past, Present and Future," 5th International Conference on Formal Ontology in Information Systems (FOIS2008), IOS Press Amsterdam, The Netherlands, Saarbrücken, Germany, 2008, pp. 3-18.
21. Vandana, K. "Ontology for Information Systems (O4IS) Design Methodology: Conceptualizing, designing and representing domain ontologies," in: *Department of Computer and Systems Sciences*, The Royal Institute of Technology, Sweden, 2007, p. 332.
22. Wand, Y., and Weber, R. "An Ontological Evaluation of Systems Analysis and Design Methods," IFIP WG 8.1 Working Conference on Information Systems Concepts: An In-Depth Analysis, Elsevier Science Publisher, Namur, Belgium, 1989, pp. 79-107.
23. Weidenhaupt, K., Pohl, K., Jarke, M., and Haumer, P. "Scenarios in System Development: Current Practice," *IEEE Software* (15:2) 1998, pp 34-45.
24. Wilson, J.M. "Gantt charts: A centenary appreciation " *European Journal of Operational Research* (2:1) 2003, pp 430-437.
25. Yu-N, C., and Abidi, S.S.R. "An Ontology-Mediated Scenario Composer for Knowledge Acquisition in Intelligent Systems," TENCON 2000, IEEE, Kuala Lumpur, Malaysia, 2000, pp. 377-381.