Application of Semantic Web Technologies in Requirements Engineering

Verónica Castañeda[1], Luciana Ballejos[1], Ma. Laura Caliusco[1], and Ma. Rosa Galli[1,2]

[1] CIDISI Research Center, UTN-FRSF, Lavaysse 610, S3004EWB Santa Fe, Argentina
vcastaneda@frsf.utn.edu.ar,{mlcaliusco,lballejos}@santafe-conicet.gov.ar
[2] INGAR-UTN-CONICET, Avellaneda 3657, S3002GJC Santa Fe, Argentina
mrgalli@santafe-conicet.gov.ar

Abstract. During the last years is increasing the interest in applying Semantic Web (SW) technologies in Requirements Engineering (RE). In this area, these technologies can be used to describe requirements specification documents, to formally represent requirements knowledge, or to formally represent domain knowledge. The purpose of this paper is to comprehensively review and present this knowledge area. The main contribution is the classification of approaches that include SW concepts within RE with the aim of clarifying the way in which the RE process can benefit from them. In addition, future trends are identified and described.

1 Introduction

The primary measure for an information system to be successful is the degree in which it meets the intended purpose. Requirements Engineering (RE) is the process of discovering that purpose by identifying stakeholders and their needs, and documenting them for their future analysis, communication, and subsequent implementation [1]. RE is understood as a subtask of Software Engineering, which proposes methods and tools to facilitate the definition of all desired goals and software functionalities.

Figure 1 shows an iterative cycle of core activities executed in a RE process [1]. All tasks presented in this figure generate diverse deliverables in order to document obtained results along the RE process. Requirements specifications, which are mainly created in the "Requirements Representation" activity, are diverse, generally complementary, and very difficult to define. Thus, software engineers are often faced with the need to redesign and iterate due to the lack of information and differences in interpretation [2].

Diverse other challenges must be faced during RE activities in order to generate consistent and complete requirements and to efficiently feed subsequent stages. One of those challenges is the management of organizations' participation (through their stakeholders) in needs and requirements gathering, considering the frequent lack of technical knowledge. Therefore, effective tools must be provided to achieve a complete analysis considering particular and general needs, and to manage requirements as a complete collaborative process [3].

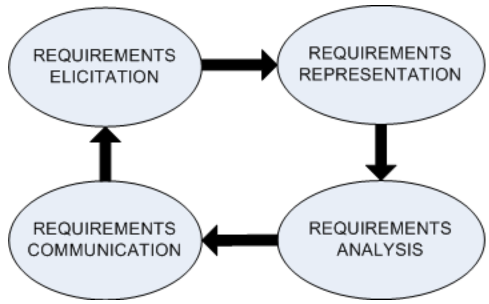

Fig. 1. *Requirements engineering activities.*

Moreover, in RE processes there is a continual need for efficiently managing the great volume of information and knowledge generated and used during all activities presented in Figure 1. Thus, ambiguous requirements must be minimized since they produce waste of time and repeated work. They arise, for example, when different stakeholders produce different interpretations for the same requirement during the "Requirements Analysis" activity.

On the other hand, determining requirements for interorganizational information systems is a major problem due to the different stakeholders' backgrounds, perspectives and individual objectives. This process may involve working with stakeholders and gathering requirements across cultural, language and time zone boundaries [4] [5] [6]. Thus, if requirements are not completely and consistently defined, the secure project planning and monitoring is in danger.

In RE, the knowledge can be associated to requirements themselves, to the variety of structures used to document them, or to the domain where the information system will be implemented and used. Then, the diverse challenges continuously faced during RE activities must be assisted by tools that help in a) the consistent requirements specification and RE documents generation, b) the representation, storage and consistent management of the great amount of knowledge generated when RE techniques and methods are applied, and c) the representation, storage and management of all the knowledge generated when analyzing the domain of interest where the system will be implemented.

With the advent of the Semantic Web (SW) and the technologies for its realization, also the possibilities for applying ontologies as a means to define the information and knowledge semantics become more and more accepted in different domains [7]. Ontologies provide a formalism to represent and relate different kind of knowledge in such a way that a machine can make deductive inferences. Related to this, the purpose of this paper is to analyze how the principal challenges in any RE process for software development are being addressed by SW initiatives.

The paper is organized as follows. Section 2 introduces the main concepts related to SW technologies. Section 3 focuses on how these technologies can help in the description of requirements specification documents and the existing approaches. Meanwhile, Section 4 describes the usefulness of SW technologies in the representation of the knowledge gathered during RE processes and Section 5 explains how they help in structuring the domain knowledge from which requirements are gathered. Moreover, Section 6 resumes the current research lines and describes the main challenges in this area. Finally, Section 7 is devoted to the conclusions of this paper.

2 Semantic Web Technologies

2.1 Ontology: Definition and Classification

The word *ontology* is rooted in the philosophy. It denotes the science of being and the descriptions for the organization, designation and categorization of existence [8]. Carried over to computer science in the fields of artificial intelligence and information technologies, an ontology is understood as a representational artifact for specifying the semantics or meaning about the information or knowledge in certain universe of discourse, in a structured form [9].

Ontologies can be classified according to the task they are meant to fulfill [10]. Knowledge representation ontologies describe the modeling primitives applicable for knowledge formalization. Top-level ontologies (also called upper-level ontologies) try to comprehensively capture knowledge about the world in general, describing for example: space, time, object, event or action and so forth, independently of a particular domain. Domain ontologies and task ontologies contain reusable vocabularies with their relations, describing a specific domain or activity. They can specialize the terms of top-level ontologies.

The ontology community also distinguishes ontologies that are mainly taxonomies from ontologies that model the domain in a deeper way and provide more restrictions on domain semantics. The community calls them *lightweight* and *heavyweight* ontologies, respectively [10].

2.2 Ontology Development Methodologies

Since ontologies have been used in different disciplines for different purposes, several methodologies for developing them have been defined [11] [12]. The objective of these methodologies is to define a strategy for identifying the key concepts that exist in a given domain, their properties and the relationships between them; identifying natural language terms to refer to such concepts, relations and attributes; and structuring domain knowledge into explicit conceptual models. Two groups of methodologies can be figured out. The first one is the group of experience-based methodologies represented by the Grüninger and Fox methodology defined in the TOVE project [13] and by the Uschold and King methodology based on the experience of developing the Enterprise Ontology [14]. The second one, is the group of methodologies that propose a set of activities to develop ontologies based on their life cycle and the prototype refinement, such as the METHONTOLOGY methodology [10] and the method defined by Noy & McGuinness [15]. Usually, the first group of methodologies is appropriate when the purposes and requirements of the ontology are clear, while the second group is useful when the environment is dynamic and difficult to understand, and the objectives are not clear from the beginning [16].

2.3 Ontology Formalization Languages

Grüber [8] proposes to model ontologies using frames and first order logic. The author have identified five kinds of components: classes, relations, functions, formal axioms and individuals. Although, with these components a high expressive power is obtained, computational properties such as decidability are not always achieved due to the reasoning complexity.

Another logical formalism for modelling ontologies is Description Logics (DL). Languages based on DL consists on two components: the TBox and the ABox [17]. The TBox describes terminology, i.e., the ontology in the form of concepts and roles definitions, while the ABox contains assertions about individuals using the terms from the ontology. Concepts describe sets of individuals and roles that describe relations between individuals.

2.4 Ontology Representation Languages: RDF and OWL

For ontology representation in a machine-interpretable way, different languages exist. The most popular are: the XML-based RDF and the formal Web Ontology Language (OWL).

RDF was originally meant to represent metadata about web resources, but it can also be used to represent information about objects that can be identified on the Web. The basic construction in RDF is an *(Object, Attribute, Value)* triplet: an object O has an attribute A with value V. A RDF-triplet corresponds to the relation that could be written as *(O, A, V)*, such as for example $(http://www.books.org/ISBN0012515866, hasPrice, 62)$, and also, for a University website annotated with an ontology $(Professor, teaches, ArtificialIntelligence)$. RDF can be used for defining lightweight ontologies.

OWL is a family of knowledge representation languages endorsed by the World Wide Web Consortium. This family of languages is based on two semantics: (1) OWL DL and OWL Lite semantics, which are based on Description Logic and have attractive and well-understood computational properties; and (2) OWL Full, which uses a semantic model in order to provide compatibility with RDF Schema.

Formal ontologies can be reasoned upon. Then, in order to allow for automated processing, any model describing concepts needs to be formal. For this purpose, description logic based OWL DL is often chosen for representing an ontology, because the more expressive OWL Full is not decidable [18]. Rules for inference can be defined based on the relationships contained in the ontology. Through inference, new knowledge can be derived through logical deduction, allowing the ontology evolution [19].

3 SW Technologies for Describing Requirements Specification Documents

A well-characterized requirements specification is important to the design stage of software development and to the evaluation and reuse of elicited requirements. Specifications are formed of both, the document structure and content. In this sense, Groza et al. [20] affirm that the structure of a document greatly influences in the perception of its content.

The use of SW technologies for describing the structure of requirements specification documents can greatly help in the definition of several structures for showing the same knowledge, in order to, for example, involve all stakeholders in the analysis of elicited requirements. Moreover, they can also help in reusing structures representation for diverse objectives or projects, only changing their content.

Reuse is one of the most required features for any software product. In order to obtain it, the form in which requirements are specified, documented and structured must be addressed. This will also help in diverse steps such as search, evaluation and adaptation, for which the existing support is insufficient. One way of exchanging reusable requirements specification documents is through Wiki systems, which allow the self-organized reuse since the community provides and organizes the artifacts to be reused [21].

The analysis of Wikis as solutions in this area is a very novel approach. The proposals conclude that requirements specification documents can specially benefit from ontologies, moreover when the content of those documents grows in a chaotic way. One way of solving this issue is structuring the knowledge by enriching the documents with additional metadata. Useful content must be found by adding semantics to the documents and extending the Wiki with RDF. In this way, the semantics is expressed in a machine-understandable format. This solution is known as Semantic Wiki and can be considered as a lightweight platform [21]. Another advantage of this approach is the automatic reasoning support and communication of the defined concepts.

Furthermore, reuse cannot be possible if requirements documents do not have two main attributes carefully balanced, as described by Hull et al. [22]: readability and processability. These two attributes can be greatly enhanced by the use of ontologies in requirements documentation. One clear example is adapted by Decker et al. [21] from the Use Case approach. They add diverse documents and new structures to the traditional Use Cases documentation. These new documents are known as templates and allow knowledge capture. Each one has metadata, besides the ontology of the documents. The authors also allow the extension of the ontology linking different Use Cases to facilitate the search of documents of the same type with other projects.

Another approach that uses templates is proposed by Groza et al. [20]. They describe a solution for generating different representations of the same document based on the metadata created by using a particular annotation framework. Proposals like this can be of great help in order to represent RE specification structures, thus promoting the reuse of RE specification content using diverse structure representations.

As mentioned before, it is widely demonstrated that the use of ontologies helps stakeholders to clarify their information needs and comes up with semantic representations of documents. Dragoni et al. [23] for example, present an approach for the ontological representation and retrieval of documents and queries for Information Retrieval Systems using a vector space model which define concepts instead of terms, where the documents are represented in a conceptual way, and the importance of each concept is calculated.

All these approaches can be, in some way, integrated in order to define an ontology for representing RE documents structures, and so, promoting the adaptation of the same content in diverse formats in order to be understandable by all stakeholders. Moreover, an ontology with this goal, can be reutilized in diverse projects in order to properly structure knowledge for each one.

4 SW Technologies to Formally Represent Requirements Knowledge

The use of ontologies for the representation of requirements knowledge has been under study since a long time ago. They help in validation and verification of requirements. They also make possible to trace dependencies among requirements, their sources and implementations.

One of the initial approaches in this area was presented by Lin et al. [24]. They propose a generic solution that provide an unambiguous, precise, reusable and easy to extend terminology with dependencies and relationships among captured and stored requirements. The proposal can be applied to any kind of product in order to reach diverse requirements: communication, traceability, completeness, and consistency. It also supports the detection of redundant or conflicting requirements. The developed ontology is implemented using Prolog. The authors propose the use of first order logic to identify the axioms and capture the definition, constraints and relationships among the objects. They also allow integrity checking of the design knowledge. Besides being a very complete proposal, one of its disadvantages is that the involved terminology is only shared by the engineers of the project, and thus, the customer is not aware of it. This way, some requirements might stand ambiguous.

The relationships among captured and stored requirements defines the traceability of the RE process. Traceability is the ability to describe and follow the life of software artifacts in Software Engineering [25]. More specifically in RE, those artifacts are the requirements. Thus, in order to trace requirements to their sources and to the intermediary and final artifacts generated from them allover the development process, it is mandatory to consider and represent information related to their source and the requirement's history.

Traceability also facilitates the reuse of the requirements and the related information. In this sense, and promoting requirements reuse, Veres et al. [26] define diverse requirement models and give rules for the mapping and traceability among them. Also Decker et al. [21] promote reuse by establishing a common requirements structure to be considered along Software Engineering activities. This is related to which Brewster et al. [27] promote, that to build systems to solve real-world tasks, not only conceptualizations must be specified, but also, clarity over the problem solving must be given.

In this way, Riechert et al. [6] present a semantic structure for capturing requirements relevant information, in order to support the RE process semantically and to promote the collaboration of all stakeholders in software development processes. They also apply and evaluate the proposal in an e-government case study.

The KAOS (Knowledge Acquisition in autOmated Specification) methodology is a goal-oriented RE approach with a rich set of formal analysis techniques [28]. KAOS is described as a multiparadigm framework that allows to combine different levels of expression and reasoning: semi-formal for modeling and structuring goals, qualitative for selection among the alternatives, and formal when needed for more accurate reasoning [29].

All goal-oriented approaches are more applicable for complex systems. They are commonly based on the not easy task of identifying goals. Then, non-functional requirements are derived from them. Their analysis and management is much more difficult than the functional requirements ones.

As a more specific approach of using ontologies for representing non-functional requirements knowledge, Dobson and Sawyer [30] propose an ontology for representing *dependability* between requirements. It considers diverse non-functional requirements, such as: availability, reliability, safety, integrity, maintainability and confidentiality.

Considering the importance of knowledge reuse and its application in RE, Wouters et al. [11] point out that one of the biggest problems in reusing use cases was to find similar or related use cases to reuse. Thus, and in order to accomplish the reuse, they propose a semiformal description which, used together with a "human" format, can make it possible the use cases reuse. The defined ontology has three categories of information: labels, concepts and relations. With these concepts diverse rules and queries can be created which, under a logic inference machine and together with algorithms, allow finding similar use cases.

From all the described proposals, the great importance of SW technologies in knowledge representation can be deduced. Moreover, in the RE area, where exist diverse points of view and the need of integrating the problem and the solution domains is latent.

5 SW Technologies to Formally Represent Domain Knowledge

Domain ontologies are specific, high-level models of knowledge underlying all things, concepts, and phenomena of a given domain of discourse. As with other models, ontologies do not represent the entire world of interest. Rather, ontologists select aspects of reality relevant to their task [31]. Then, the selection of the methodology to be used for developing an ontology depends on the application that ontologists have in mind and the extensions that they anticipate.

In software development, an ontology can be used at development time or at run time [32]. Using an ontology during the development stage enables designers to practice a higher level of knowledge reuse that is usually the case in software engineering. At run time, an ontology may enable, for instance, the communication between software agents or be used to support information integration. In both cases, the creation of the ontology starts at the RE process.

Any software development process implies multiple stakeholders which collaborate with a common goal. At development time, a domain ontology can be used as a way of facilitating the understanding between stakeholders. Pohl [33] affirms that RE must elicit and understand the requirements from the relevant stakeholders and develop the requirements together with them. Thus, in order to maximize environment comprehension, a common understanding of the involved concepts must be carried out. This means, the requirements analysts should be endeavored and must work towards understanding the language used in the universe of discourse, to then initiate the modeling of that universe. A model of the environment represents the reality and considerably improves its comprehension. Thus, a crucial part of RE is the establishment of a common terminology by diverse stakeholders. To this aim, the methodologies described in Section 2.1 can be used at the first stage of the software development process.

The traditional methodologies for development ontologies appear to be unusable in distributed and decentralized settings, and so the systems that depend on them will fail to cope with dynamic requirements of big or open user groups [34]. In this sense, Breitman and Sampaio do Prado Leite [35] propose a process for building an application ontology during the requirements process based on the Language Extended Lexicon (LEL). The lexicon provides a systematization for the elicitation, model and analysis of ontology terms. The underlying philosophy of the lexicon falls in the *contextualism* category, according to which particularities of a system's context of use must be understood in detail before requirements can be derived. This approach is new to ontology building, which traditionally associates generalization and abstraction approaches to the information organization. Application ontologies are much more restricted than domain ontologies and have a much more modest objective. The authors see the ontology of a web application as a sub-product of the RE activity.

6 Semantic based Requirements Engineering

The approaches previously explained integrate ontologies to RE working over the basis of three well demarcated lines, in an isolated form:

- **Requirements structure ontology generation**: represent terminology used in requirements traditional documents [21] [36],
- **Requirements content ontology generation**: represent knowledge arrived at by consensus [24] [11] [35],

– **Requirements domain ontology generation**: represent the meaning of the terminology used in RE activities [30] [37].

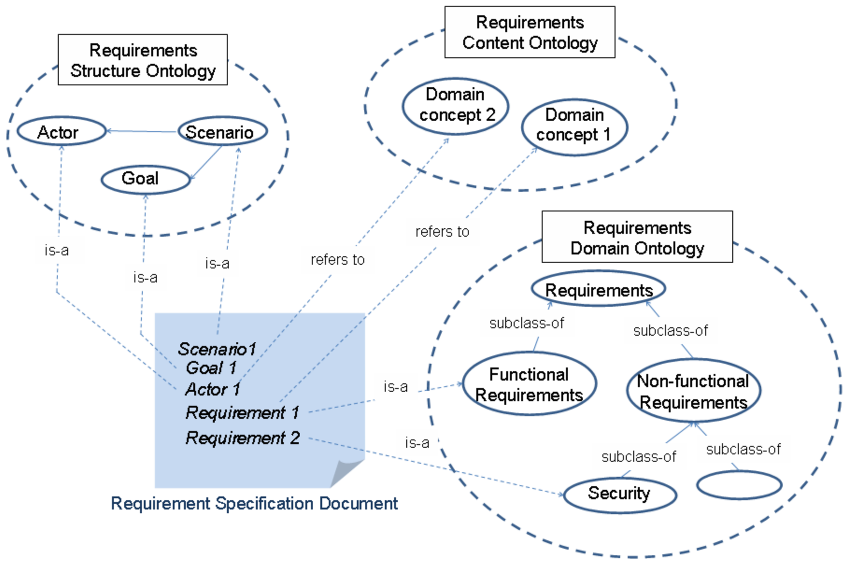

Fig. 2. *Semantic based Requirements Engineering.*

Figure 2 shows the relation among these ontologies and the knowledge contained in the requirements specification documents. These ontologies contribute to minimize different obstacles presented in diverse tasks involved in RE that come from a misunderstanding due to a wrong conceptualization:

– **Ambiguous Requirements**: which produce waste of time and repeated work. Their origin resides in the diverse stakeholders, who produce different interpretations of the same requirement. Moreover, one stakeholder can interpret the same requirements in diverse ways. The ambiguity conduce to mistaken product tests. To reduce this obstacle, a requirements domain ontology can be used during the elicitation process.
– **Insufficient Specifications**: they produce the absence of key requirements. This conduce to developers frustration, because they base their work in incorrect suppositions and, so, the required product is not developed, which displeases the clients. A requirements domain ontology and a requirements structure ontology could be used to reduce this obstacle.
– **Requirements not completely defined**: they make impossible the project secure planning and its monitoring. The poor understanding of requirements leads to optimistic estimations, which return against when the agreed limits

are surpassed. To reduce this obstacle, a requirements domain ontology can be used during the elicitation process.
– **Dynamic, changing requirements**: which require constant requirements revision in order to help to understand new clients' needs and to identify how they can be satisfied. A requirement content ontology is useful to identify dynamic and changing requirements.

The application of SW technologies is beneficial for RE processes, since a crucial part of RE is the establishment of a common terminology by different (often spatially distributed) stakeholders. However, in order to obtain the maximal benefit it is necessary to define a framework to support the collaboration of all stakeholders in the definition of requirements along all involved tasks, and moreover, to define a common structure and knowledge representation format.

Thus, in order to integrate the three areas previously described, the use of SW technologies can be of great help. They could facilitate and materialize the reuse of the knowledge generated and managed allover the software development project lifecycle.

7 Conclusions

This paper describes diverse trends in which SW technologies are applied in several areas addressed by RE. The described approaches are very specific proposals: they can be applied in certain circumstances and activities of RE. From this, the latent need of an integrative approach can be determined.

The application of SW technologies is beneficial for RE processes. Moreover, several are the topics and areas in RE in which ontologies can be of great help. The three main of them were pointed out in this paper. So far, there is not any approach to address these challenges in an integrated way. To solve this issue the future work will be focused on defining a framework to support the requirements structure ontology generation, the requirements content ontology generation and the requirements domain ontology generation. This will allow the collaboration of all stakeholders in the definition of requirements along all involved tasks in RE, and moreover, to define a common structure and knowledge representation format for elicited requirements, minimizing the occurrence of diverse challenges.

This framework will be useful in requirements consistent management, requirements specification, and requirements knowledge representation activities during the entire software development project.

References

1. Nuseibeh, B., Easterbrook, S.: Requirements engineering: A roadmap. In: International Conference on Software Engineering - Conference on the Future of Software Engineering. (2000) 35–46
2. Noppen, J., van den Broek, P., Aksit, M.: Imperfect requirements in software development. In P. Sawyer, B.P., (Eds.), P.H., eds.: Requirements Engineering

for Software Quality - REFSQ 2007. Volume LNCS 4542., Springer-Verlag Berlin Heidelberg (2007) 247–261
3. Lang, M., Duggan, J.: A tool to support collaborative software requirements management. Requirements Engineering Journal **6** (2001) 161–172
4. Ballejos, L., Montagna, J.: Method for stakeholder identification in interorganizational environments. Requirements Engineering **13**(4) (2008) 281–297
5. Damian, D., Zowghi, D.: Requirements engineering challenges in multi-site software development organizations. Requirements Engineering Journal **8**(3) (2003) 149–160
6. Riechert, T., Lauenroth, K., Lehmann, J., Auer, S.: Towards semantic based requirements engineering. In: Proceedings of the 7th International Conference on Knowledge Management (I-KNOW). (2007)
7. Shadbolt, W., Berners-Lee, T.: The semantic web revisted. IEEE Intelligent Systems **21**(3) (2006) 96–101
8. Grüber, T.: A translation approach to portable ontology specification. Knowledge Acquisition **5**(2) (1993) 199–220
9. Allemang, D., Hendler, J.A.: Semantic web for the working ontologist: Modeling in RDF, RDFS and OWL. Elsevier, Amsterdam (2008)
10. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Springer, New York. USA. (2004)
11. Wouters, B., Deridder, D., Van Paesschen, E.: The use of ontologies as a backbone for use case management. In: Proceedings of European Conference on Object-Oriented Programming (ECOOP 2000). (2000)
12. Corcho, O., Fernández-López, M., Gómez Pérez, A.: Methodologies, tools and languages for building ontologies. where is the meeting point? Data & Knowledge Engineering **46** (2003) 41–64
13. Grüninger, M., Fox, M.: Methodology for the design and evaluation of ontologies. In: Proceedings of the Workshop on Basic Ontological in Knowledge Sharing. (1995)
14. Uschold, M.: Building ontologies: towards a unified methodology. In: Proceedings of 16th Annual Conference of the British Computer Society Specialists Group on Expert Systems. (1996)
15. Noy, N., McGuinness, D.: Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics (2001)
16. Cristiani, M., Cuel, R.: A comprehensive guideline for building a domain ontology from scratch. In: Proceedings of International Conference on Knowledge Management (I-KNOW '04). (2004)
17. Baader, F., M.D.N.D., Patel-Scheneider, P.: The Description Logics Handbook: Theory, implementation and applications. Cambridge University Press, Cambridge, United Kingdom (2003)
18. Smith, M., Welty, C., McGuinness, D.: Owl web ontology language guide. W3C recommendation **10** (2004)
19. Antoniou, G., Franconi, E., van Harmelen, F.: Introduction to semantic web ontology languages. Reasoning Web **3564**(1) (2005) 1–21
20. Groza, T., Schutz, A., Handschuh, S.: Salt: A semantic approach for generating document representations. In: Proceedings of the 2007 ACM symposium on Document engineering. (2007) 171–173
21. Decker, B., Rech, J., Klein, B., Hoecht, C.: Selforganized reuse of software engineering knowledge supported by semantic wikis. In: Proceedings of Workshop on Semantic Web Enabled Software Engineering (SWESE). (2005)

22. Hull, M., Jackson, K., Dick, A.: Requirements Engineering, Practitioner Series. Springer, New York. USA. (2002)
23. Dragoni, M., da Costa Pereira, C., Tettamanzi, A.: An ontological representation of documents and queries for information retrieval systems. In: Proceedings of the 1st Italian Information Retrieval Workshop (IIR '10). (2010)
24. Lin, J., Fox, M., Bilgic, T.: A requirement ontology for engineering design. Manuscript, Enterprise Integration Laboratory, University of Toronto (1996)
25. Winkler, S., von Pilgrim, J.: A survey of traceability in requirements engineering and model-driven development. Software and Systems Modeling **Theme Section** (2009)
26. Veres, C., Sampson, J., Bleistein, S., Cox, K., Verner, J.: Using semantic technologies to enhance a requirements engineering approach for alignment of it with business strategy. In: Proceedings of 2009 International Conference on Complex, Intelligent and Software Intensive Systems. (2009)
27. Brewster, C., O'Hara, K., Fuller, S., Wilks, Y., F.E., Musen, M., Ellman, J., Buckingham Shum, S.: Knowledge representation with ontologies: The present and future. IEEE Intelligent Systems **19**(1) (January/February 2004) 72–81
28. Lapouchnian, A.: Goal-oriented requirements engineering:an overv iew of the current research. Technical report, University of Toronto, Department of Computer Science (2005)
29. van Lamsweerde, A., Letier, E.: From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. In: Radical Innovations of Software and Systems Engineering in the Future. Volume 2941 of Lecture Notes in Computer Sciences. Springer Berlin / Heidelberg (2004) 153–166
30. Dobson, G., Sawyer, P.: Revisiting ontology-based requirements engineering in the age of the semantic web. In: Proceedings of the International Seminar on Dependable Requirements Engineering of Computerised Systems. (2006)
31. Devedzic, D.: Understanding ontological engineering. Communications of the ACM **45**(4) (2002) 136–144
32. Fonseca, F.: The double role of ontologies in information science. American Society for Information Science and Technology **58**(6) (2007) 786–793
33. Pohl, K.: Requirements Engineering. Dpunkt Verlag (2007)
34. Davies, J., S.R., Warren, P.: Semantic Web Technologies: Trends and research in ontology-based systems. John Wiley, England (2006)
35. Koogan Breitman, K., Sampaio do Prado Leite, J.: Ontology as a requirements engineering product. In: Proceedings of the 11th IEEE International Requirements Engineering Conference. (2003)
36. Mayank, V., Kositsyna, N., Austin, M.: Requirements engineering and the semantic web, part ii. representation, management, and validation of requirements and system-level architectures. Technical report, University of Maryland (2004)
37. Jureta, I., Mylopoulos, J., Faulkner, S.: Revisiting the core ontology and problem in requirements engineering. In: Proceedings of the 16th IEEE International Requirements Engineering Conference, 2008 (RE'08). Best Paper Award. (2008)