Vol-596
urn:nbn:de:0074-596-3

ORES-2010
Ontology Repositories and Editors for the Semantic Web

Proceedings of the 1st Workshop on Ontology Repositories and Editors for the Semantic Web

Hersonissos, Crete, Greece, May 31st, 2010.

Edited by

Mathieu d'Aquin, The Open University, UK
Alexander García Castro, Universität Bremen, Germany
Christoph Lange, Jacobs University Bremen, Germany
Kim Viljanen, Aalto University, Helsinki, Finland

10-Jun-2010: submitted by Christoph Lange
11-Jun-2010: published on CEUR-WS.org

Evaluation Framework for Ontology Development and Management Methodologies

Dionisis D. Kehagias, Dionysia Kontotasiou, and Dimitrios Tzovaras

Centre for Research and Technology Hellas,
Informatics and Telematics Institute,
57 001, Thermi, Greece,
diok@iti.gr, dkonto@iti.gr, dimitrios.tzovaras@iti.gr

Abstract. Based on the IEEE 1074-1995 standard for software development process we define a set of complement criteria for the evaluation of existing ontology development and management methodologies. We use this evaluation framework in order to benchmark the most well known approaches for developing ontologies from scratch, as well as reusing ontologies that are stored in ontology repositories. The result of the evaluation process is to identify the shortcomings of existing ontology development and management methodologies and validate the use of the aforementioned criteria for the establishment of a generic evaluation framework. Moreover, the conducted evaluation procedure reveals the degree of conformance of the benchmarked methodologies to a set of standardized criteria. In order to include the newly introduced evaluation criteria whose purpose is to support more ontology-specific evaluation aspects, we extend the evaluation framework introduced in the Ontoweb project by also adding nine additional methodologies.

1 Introduction

Ontology development and management are two knowledge engineering processes that are required either for constructing an ontology or an ontology network from scratch or reusing existing ontologies. Even though many visualization support tools that are available today facilitate the various steps of the ontology lifecycle, the core development of an ontology remains a manual task that requires good knowledge of the domain to be modeled, as well as good modeling skills and experience. To this end a deep knowledge of existing methodologies about ontology management and development are necessary for the adoption of the best practices available in the market.

This paper aims at the provision of an in-depth review of the most common methodologies for ontology development and management by introducing a set of evaluation criteria to enable benchmarking of the existing methodologies. The proposed set of criteria is based on the IEEE 1074-1995 standard for software development process [1] that defines the main processes involved in the administration, preparation, development and integration of a software project. Various ontology construction methods for building single ontologies from scratch are presented in [2], where it is noted that none of them is the most adequate for all situations; instead, each one has its own use, depending on the application's specificity. Our work extends the IEEE 1074-1995 in order to form the basis for the development of a concrete methodological approach and a set of guidelines for ontology authoring which can be used and applied as a set of evaluation criteria for new ontologies as well.

In particular the paper focuses on the introduced criteria by providing a thorough analysis of their impact on existing methodologies. The new evaluation criteria are based on practical observation of ontologies from real repositories and a set of practical guidelines about the establishment of an ontology evaluation and refinement best practice. By extending the IEEE standard we conduct benchmarking of existing ontology development and management methodologies in order to identify potential inefficiencies on existing methodologies. Our evaluation framework extends the one introduced by the Ontoweb project [3] by adding nine additional methodologies. An evaluation process is conducted in order to investigate the degree in which the existing methodologies provide support of the evaluation criteria presented in this paper.

2 The IEEE Software Development Standard

The set of criteria used for the definition of our evaluation framework is based on the IEEE 1074-1995 standard for software development process. According to the IEEE definition [1], software is ”computer programs, procedures, and possibly associated documentation concerned with the operation of a data processing system; e.g. compilers, library routines, manuals, and circuit diagrams”; ontologies are part of software products. In order to construct an ontology similar processes are required, such as design of class hierarchies, development of the ontology, validation of an ontology by executing a reasoner and generation of documentation. Based on these similarities, the proceeses involved to the creation of an ontology from its conceptualization to the development and documentation can be described by the IEEE standard after its adaptation to the specific characteristics of ontology development.

According to this standard any software development process is broken down in processes. We adapt each process to existing ontology development methodologies and propose a set of applied criteria. The purpose of the set of defined criteria is to drive the evaluation procedure for different ontology development and management methodologies. These criteria are intended to stand as complementary elements to the IEEE 1074-1995 standard extending its application to the ontology development and management processes. The IEEE 1074-1995 standard defines the following processes for software.

- *Project management processes*. These processes adhere to the procedure required for setting up a software development project. Their purpose is to ensure the right level of management throughout the entire project life cycle. They include activities related to project initiation (such as participants, scheduling, etc.), project monitoring and control, and quality management. The activities proposed by the IEEE standard for these processes are applicable to any software product and therefore they are recommendable to be applied in ontology development. Ontology management activities include scheduling, control and quality management. The scheduling activity identifies the tasks to be performed, their arrangement, and the time and resources needed for their completion. This activity is essential for ontologies that use ontologies stored in ontology libraries or for ontologies that require a high level of abstraction and generality. The control activity guarantees that scheduled tasks are completed in the manner intended to be performed. Finally, the quality management activity assures that the quality of each and every product output (ontology, software and documentation) is satisfactory.

- *Development-oriented processes*. This category includes the processes that are used in order to produce, install, operate and maintain the software and retire it from use. They are divided into three groups:
    a. *Pre-development processes*. They are performed prior to the actual software development. They involve activities related to the study of the software installation environment, and to feasibility studies. During the ontology pre-development an environment study identifies the problem to be solved with the ontology, the applications that will consume the ontology, etc. Also during the pre-development, the feasibility study answers questions such as: ”what is the purpose for building such and ontology”; ”is this ontology suitable to solve the problem for which it is designed?”
    b. *Development processes*. These are the required processes for building the software product. They include: requirements, which are comprised of iterative activities directed towards developing the software requirements specification; the design process, the goal of which is to develop a coherent and well-organized representation of the software system that meets the requirements specification; and the implementation process, which transforms the design representation of a software product into an implementation language. Obviously, if ontologies are to be used by computers, they have to be implemented like software products. Thus, firstly in development, the specification activity states why the ontology is being built, what are its intended uses and who are the end-users. The conceptualization activity, like software design process, structures the domain knowledge as meaningful models at the knowledge level either from scratch or by reusing existing models. Finally, the implementation activity builds computable models in an ontology language.
    c. *Post-development processes*. They are related to the installation, operation, support, maintenance and retirement of a software product. They are executed after the software construction. As in software, these activities are applied to ontologies in the way we explain in what follows. During post-development maintenance activities concern updates and corrects the ontology if needed. Also during post-development, the ontology is (re)used by other ontologies or applications. Evolution involves managing ontology changes and the impact of updated versions of the ontology, taking into account the applications and the environments on which it can be used.
- *Integral processes*. These processes are required for successful completion of software project activities. They are executed concurrently to the software development-oriented processes and include those activities that are necessary for the successful integration of the overall system. With respect to software development and management they cover the processes of knowledge acquisition, verification and validation, software configuration management, documentation development and training. The activities proposed by the standard for these processes can be applied

also to ontologies in the following ways. Ontology evaluation involves assessment of ontologies and associated execution environments from a technical point of view. Documentation for ontologies is as necessary as in software products. Configuration management can be applied to ontology as a means of assessment to make sure that the developed ontology adheres to its original requirements.

After we have seen how the IEEE Standard can be applied to ontology development, in the next Section we describe the complement criteria that we have introduced to the above processes defined in the IEEE standard in order to establish an ontology management evaluation framework.

3 Evaluation Criteria

We present here the additional ontology evaluation criteria that we have defined with respect to the aforementioned IEEE standard in order to make the standard applicable to the ontology development and management processes. For presentation purposes we introduce the new criteria based on practical experience by the application of the aforementioned process categories that are defined in the IEEE standard on existing ontology repositories. Such a repository on which we have been experimented is the ORATE (Ontology Repository of Assistive Technologies), which is located at `http://ontologies.informatik.uni-bremen.de/`. ORATE hosts a large collection of assistive-related ontologies.

3.1 Project Management Processes

Project management includes on-going activities that are executed during the whole period of the ontology capture and development process. We do not provide new criteria for this category. However we review the most common methodologies for ontology management processes in general with respect to their support for project management activities.

3.2 Pre-Development Processes

During the ontology pre-development phase an environment study identifies the problem to be solved with the ontology, the applications where the ontology will be integrated, etc. Also the feasibility study provides a mechanism to refine the vision statement and to find out whether an ontology is actually worthwhile in terms of expected costs and benefits. A feasibility report not only provides recommendations of how to re-fine the vision statement, but also the material and rationale underpinning it. As previously, we do not provide new criteria for this category. However we review the most common methodologies according to their support for these pre-development processes.

3.3 Development Processes

The development processes category includes most of the new criteria. Firstly, in this process category that involves those processes that are executed in order to prepare the main development process, we introduce two new criteria, namely *concept hierarchy* and *property structure*. Hierarchy renders a key aspect in the ontology development process. Since hierarchy of the various concepts is part of the ontology design process and involves decisions made at the initialization of the ontology development phase it is related to development processes. More specifically it is related to ontology conceptualization. Evaluation metrics that are derived from this criterion are the size, the depth, and the breadth of hierarchy, the density (average branching of concepts), etc., which define the overall complexity of the ontology. A flat concept hierarchy for example usually means that there are too many concepts on the same level [4]. This phenomenon implies the existence of unexploited grouping possibilities for concepts of similar kinds, e.g. to be grouped together under one more general concept. Another example is the existence of branches very differently structured than others (e.g. very big depth), something that results in an unbalanced taxonomy. In general, if the level of abstraction to which the concepts refer is not taken into careful consideration, the result will be an inappropriate design of the ontology.

Next criterion is the *property structure*. This criterion, like the previous one, refers to development processes, as the definition of an appropriate property structure is realized in the development of an ontology. This criterion is associated with metrics such as the size, the depth/breadth of hierarchy, density and complexity of the ontology, etc. It is often observed in ontologies for which data or object properties are not properly structured or not structured at all. In this case, a restructuring process might be necessary by exploiting grouping possibilities for properties of equal domains/ranges or their functions.

Two more criteria in the same category are *domain/range definition of properties* and *disjointness restrictions*. The domain/range definition criterion covers the activity of defining the environment to which the ontology has impact. For this reason and based on the IEEE standard we classify this as a development process. The existence of properties which do not define their domain/range can cause significant inconsistencies when using the ontology. Another common case is the existence of object properties which do not define their range, but instead they appear in restrictions of concepts, in which the range is set (as a condition of the concept).

Next criterion, disjointness restrictions [5] is also implemented as a development process. Its impact is visible when the ontology is used as part of an overall application (e.g. when instances are added, forms are created, queries have to be managed, etc.). Although most concepts inside the ontology are usually pairwise disjoint with each other, this condition is not always there. On the other hand, for some other concepts disjointness should not hold when there exists an individual that is an instance of two classes. In this case disjointness restriction should be removed from the two classes.

3.4 Post-Development Processes

Three ontology-specific criteria are introduced in the post-development process category, that correspond to activities such as support, maintenance and retirement. These are *repetition of similar ontological concepts*, *subtraction of modules* and *naming conventions*.

The first criterion is presented in the post-development processes category because it is related to the main ontology post-development activities of maintenance and reuse. In particular this criterion concerns modularization (e.g. what modules are defined in the ontology, how they are defined, if they can be imported/exported/reused, etc). If similar ontological concepts are repeated frequently throughout the ontology structure, they can possibly be combined to one module and reused whenever necessary. Hence, repeated concepts can be defined only once and their use be extended within other definitions.

The second criterion, subtraction of modules, is closely related to the previous one, since it refers to subtracting modules in general (either functional or logical) from the whole ontology, which is also the result of applying measures in order to eliminate repetition of similar concepts. Such an action can reduce the overall complexity and elucidate dependencies between various ontology parts. From our practical experience with ORATE ontologies we noticed that, some ontologies have duplicate definitions of the same concept or concepts which are very similar (or almost identical to each other). In such a case, it is necessary to eliminate duplicate definitions and remove similar concepts or merge them to a single one. Moreover, there might be properties initially created for some purpose, but finally never used at all. These properties should also be removed. All these steps usually result in an ontology of reduced complexity, more ”clear”, compact and readable.

Moreover, in this category we have included a criterion about naming conventions [6]. This criterion has to do with the formulation of ”good” terms and definitions, where essential features should be satisfied by all naming conventions (e.g. nominal, verbal, etc). Circularity in definitions should be avoided and junk categories should be eliminated.

3.5 Integral Processes

In the group of integral processes we have included a criterion about documentation and information visualization. The integral processes include the activities of validation and documentation development. They are related to documentation, syntax (syntactic correctness, breadth of syntax used), and governance in used terms, etc. The specific criterion concerns the activity of enriching the ontology with additional information (e.g. natural language comments/annotations, metadata, implementation code, etc.), as well as the collection of documents and explanatory comments generated during the entire ontology building process. In general, this issue has to do with anything that could be useful to help users, who did not participate in the ontology development, to understand and learn how the ontology was built.

4 Evaluation of Existing Ontology Development Methodologies

This section presents a survey of ontology development methodologies and the results obtained after conducting benchmarking of the existing tools with respect to our IEEE

1074-1995 standard-based evaluation framework. A short description of the main characteristics of the ontology development and management methodologies that participated in our benchmarking evaluation is provided in what follows.

4.1 Ontology Development and Management Methodologies

The *Cyc* methodology which arose from experience of the development of the *Cyc* knowledge base contains a huge amount of common sense knowledge [5]. After evaluating *Cyc* we saw that it provides limited description of the criteria and processes described in Section 3. For example, the criteria for formulating the concept hierarchy are not mentioned.

The *Uschold and King* ontology development method [7], also known as the "skeletal method", is based on the experience of developing the Enterprise Ontology which is a collection of terms and definitions relevant to business enterprises. This method is composed of four distinct stages: identification, construction, evaluation and documentation. However, some criteria are missing in the processes it does propose (development and integral), particularly: concept hierarchy, property structure, naming conventions and information visualization.

The *Toronto Virtual Enterprise Method* (TOVE) [8] was derived from the authors' own experience in developing ontologies for business and corporate processes, using motivating scenarios to describe problems and examples that were not addressed by existing ontologies. This methodology is very formal and can be used as a guide to transform informal scenarios in computable models. However, it shows similar omissions as the previous methodology. In particular, no reference is made to the criteria concerning: naming conventions and documentation but it provides more details for the criteria in the processes it does propose, i.e. development and post-development processes. These include concept hierarchy, disjointness restrictions and subtraction of modules.

The *METHONTOLOGY* framework [9] is essentially a descriptive method that provides automated support for ontology development and is based on the IEEE 1074-1995 standard for software development; it suggests which criteria should be accomplished when building ontologies, but it does not provide guidance as to how they should be carried out. Thus, some activities and techniques relevant to the post-development processes should be specified in more detail.

The method based on *SENSUS* [10] is completely different from the others. Domain ontologies built using the *SENSUS* approach share the same high level concepts (or skeleton). Thus systems that use such ontologies will share a common structure of the world, and it would be easier for them to communicate because they share the same underlying knowledge. However, this methodology does not mention at all any post-development processes that are required in order to ensure that the resulted ontologies satisfy a set of usability standards.

On the contrary, *CommonKADS* methodology [11] does not put emphasis on management and integral processes but only to pre-development and development processes. According to this methodology, the phases to ontology design are: feasibility study, refinement and evaluation, while the management processes are missing.

The partially well-documented *On-To-Knowledge* methodology [12] includes the identification of goals that should be achieved by knowledge management tools and it is based on an analysis of usage scenarios. It is a centralized ontology development method that risks becoming too much geared towards a single application and not towards satisfying general management and development criteria. This could also be a potential problem for *ROD* [13] which is a methodology that is used to build ontologies for under developed domains without ensuring that the resulted ontologies satisfy a set of usability standards. It consists of three processes: domain analysis, document and language processing which correspond to development processes.

Holsapple et al. [14] focus their methodology on the collaborative aspects of ontology engineering but still aim at a static ontology. According to their methodology a knowledge engineer defines an initial ontology which is extended and modified based on the feedback from a panel from domain experts. This feedback does not include criteria relevant to the post-development and integral processes.

The *DOGMA* methodology [15] is quite similar to *DILIGENT* [16] and *HCOME* [17] methodologies. All these efforts move towards the third-generation of ontology engineering methodologies. Specifically, they focus on management and development criteria and thus emphasize on issues concerning good representation and architecture of the ontologies. In addition, these methodologies consider evolving ontologies, pointing on the importance of documentation, versions management and merging of ontologies. All three of them omit the post-development processes criteria.

UPON (Unified Process for ONtology building) [18] is an incremental methodology for building ontologies. This methodology stems its characteristics from the Software Development Unified Process and uses the Unified Modeling Language (UML) to support the preparation of all the blueprints of the ontology project. Because of its nature, *UPON* does not deal with management and integral issues. On the other side it describe in detail the development criteria, which is an advantage over the adoption of other methodologies, that roughly cover the same criteria as *UPON*.

Karapiperis and Apostolou [19] proposed a methodology which complies almost perfectly with our criteria. This approach starts with the deployment of an initial version of the ontology, created by the coordinator, based on the participants' requirements. The initial version is being iteratively evaluated by the participants and it finally evolves into the final version. It ensures that all participants agree and accept the resulting ontology, being a product of a joint team effort. These phases comply also with Holsapple's phases [14]. However, due to the iterative cycles of the consensus building mechanism the collaborative ontology approach require more time and effort to deployment as opposed to other approaches.

In [20] a novel modeling methodology for biomedical ontologies is designed called GM. This methodology has the similar compliance to our evaluation framework as the above methodology. A key feature of this methodology is the use of Concept Maps (graphs consisting of nodes representing concepts, connected by arcs representing the relationships between those nodes) throughout the knowledge acquisition process. Unlike GM, *iCapturer* [21] makes use of all the development criteria except from concept hierarchy and property structure. This methodology does not include the applicability of an ontology in a given application domain.

Last but not least *NeOn* (http://www.neon-project.org) is a framework for developing networked ontologies. It is one of the most comprehensive works in terms of ontology engineering. The framework incorporates a methodology. The first version of the *NeOn* Methodology for collaboratively building networks of ontologies is available since February 2008. This version of the methodology is focused on the post-development and integral processes. The second version of the NeOn Methodology [22] for collaboratively building networks of ontologies launched in February 2009.

4.2 Benchmarking Results

The result of the comparative evaluation of the aforementioned key methodologies is illustrated in Table 1 where the conformance of each methodology to our evaluation criteria is shown. Rows in Table 1 represent the various methodologies, while columns represent the different groups of processes as they are defined in the IEEE 1074-1995 standard. Each cell in the table can be filled in with five types of values. The value ”described” (D) means that the approach establishes for the considered metric: how to do each task, when to do it, who has to do it, etc. The value ”proposed but not described” (P-ND) means that the methodology of the corresponding row identifies the process that is written in the column as a process to be performed during the ontology development process but there is no description for this process. The value ”not proposed” (NP) means that public documentation does not mention the non-considered aspect. Finally ”limited” (L) means that limited information is provided for the particular group of processes.

According to this table, there is no methodology with a full conformance to our criteria except from *Consensus-based* and GM methodologies. However the *Consensus-based* ontology approach to ontology engineering may require more time and effort to deployment as opposed to other approaches due to the iterative cycles of the consensus building mechanism but this tradeoff is expected to be improved in the long term. In addition, the GM methodology emphasizes the notion of collaboration in the development process, particularly during knowledge acquisition. The GM knowledge acquisition relies heavily on interaction; the higher the level of interaction amongst domain experts, the more refined the specific models are likely to be.

The purpose of the benchmarking process we conducted was to figure out how sufficiently each methodology conforms to the proposed evaluation criteria. The process adopted for this purpose was similar to the one adopted in the *OntoWeb* project. A description of methodologies for developing, maintaining, evaluating and re-engineering ontologies is provided in the public deliverable D1.4 of the *OntoWeb* project [3], while a thorough survey can be also found in [23]. Our evaluation procedure presented in this paper extended the one used in *OntoWeb* by adding nine more methodologies to the existing ones.

5 Conclusions and Future Work

In this paper we presented a comparative analysis whose goal is to benchmark a set of different ontology management and development methodologies according to a set of criteria. For the deployment of our evaluation framework we have used the one adopted by the *Ontoweb* project as a basis. Because there are quite a few survey papers on

Table 1. Comparison of various ontology development methodologies with respect to the new criteria based on the IEEE 1074-1995 standard processes. The following abbreviations are used. NP: not proposed, D: described, L: limited, ND: not described, P: proposed.

	Processes distinguished in the IEEE 1074-1995 standard and corresponding criteria				
Methodologies	**Project management**	**Pre-development**	**Development**	**Post-development**	**Integral**
Cyc	NP	NP	NP	NP	L
Uschold & King's	ND	ND	L	NP	L
TOVE	ND	ND	D	NP	L
METH-ONTOLOGY	D	D	D	P-ND	D
KACTUS	D	D	D	NP	L
SENSUS	D	D	D	NP	D
CommonKADS	L	D	D	NP	L
OTK	L	D	L	NP	L
ROD	D	D	D	D	L
Holsapple	D	D	D	NP	L
DOGMA	D	D	D	NP	D
DILIGENT	NP	D	NP	L	D
HCOME	D	D	D	P-ND	D
UPON	NP	D	D	NP	L
Consensus-based	D	D	D	D	D
GM	D	D	D	D	D
iCapturer	D	P-ND	D	D	D
NeOn	NP	L	NP	D	D

methodology, we provided a brief description of the most well known approaches for building ontologies both from scratch, or reusing ontologies from existing ontology repositories. Finally we evaluated them according to the set of criteria proposed based on the IEEE 1074-1995 standard.

Moreover, our framework extends the *Ontoweb* framework by also including nine additional methodologies. The evaluation results reveal conformance of the evaluated methodologies according to the different criteria proposed in this paper. We have seen that Consensus-based and GM methodologies comply almost perfectly with our criteria. However, *Consensus-based* and GM are difficult to be applied. *Consensus-based* ontology approach requires more time and effort to deployment due to the iterative cycles of the consensus building mechanism. In addition, the GM methodology emphasizes the notion of collaboration in the development process, particularly during knowledge acquisition. The fact that GM knowledge acquisition relies heavily on interaction makes this methodology difficult to use and understand, especially for users who want to apply/reuse it (for example, for their own application). Therefore, a solution that balances between performance, usability and conformance to criteria should be mainly sought among *SENSUS*, *METHONTOLOGY*, *DILLIGENT* and *HCOME*.

The comparative analysis proposed in this paper extended the Ontoweb framework that has been used as evaluation framework for ontology development methodologies by adding further criteria that drive the comparison. These criteria support domain experts, users, knowledge engineers and ontology engineers in collaboratively restructuring a shared ontology. Moreover, the presented evaluation framework guides the participants in a perfect way through the ontology development lifecycle, allowing for personalization and taking into account specific criteria as the proposed ones. Our future plans include the establishment of quantitative metrics to measure the conformance of the benchmarked methodologies with respect to the different evaluation criteria.

A real case of how the aforementioned criteria can be applied on a real ontology is going to be presented. The described process will be applied to a set of existing ontologies. The expected outcome of the presented case study and evaluation will form the basis for the development of a concrete methodological approach and a set of criteria for ontology evaluation.

References

1. IEEE Standard for Developing Software Life Cycle Processes. IEEE Computer Society, New York (USA). April 26, 1996.
2. K.K. Breitman, M.A. Casanova and W. Truszkowski, ”Methods for Ontology Development”. Semantic Web: Concepts, Technologies and Applications, Part III, Chapter 8, pp. 155-173, Springer, 2007.
3. OntoWeb Project IST-2000-29243, Deliverable 1.4: ”A Survey on Methodologies for Developing, Maintain-ing, Evaluating and Reengineering Ontologies”, Available online: http://www.ontoweb.org/About/Deliverables/OverviewProjectPhase3/D1.4-v1.0.pdf
4. M. Delanda, Intensive science and virtual philosophy, Continuum, New York, 2002.
5. D.B. Lenat and R.V. Guha, ”Building large knowledge-based systems”. Addison-Wesley Publising Com-pany, Inc. 1990.
6. A.L. Rector, ”Modularisation of domain ontologies implemented in description logics and related formal-isms including OWL”. In Proceedings of the 2nd international Conference on Knowledge Capture (Sanibel Island, FL, USA, October 23 - 25, 2003).
7. M. Uschold and M. King, ”Towards a Methodology for Building Ontologies”. In Proc. of Workshop on Basic Ontological Issues in Knowledge Sharing (held in conjunction with IJCAI’95), 1995, Montreal, Can-ada.
8. M. Gruninger and M. Fox, ”Methodology for the Design and Evaluation of Ontologies”. In Proc. of Work-shop on Basic Ontological Issues in Knowledge Sharing (held in conjunction with IJCAI’95), 1995, Mont-real, Canada.
9. M. Fernández-López, A. Gómez-Pérez and N. Juristo, ”METHONTOLOGY: From Ontological Art Towards Ontological Engineering”. In Proc. of Spring Symposium on Ontological Engineering (AAAI’97), pp. 33-40, 1997, Stanford, CA, USA.
10. B. Swartout P. Ramesh, K. Knight and T. Russ, ”Toward Distributed Use of Large-Scale Ontologies”. In Proc. of Spring Symposium on Ontological Engineering of AAAI, pp. 138-148, March 24-26, 1997, Stan-ford, CA, USA.
11. G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde and B. Wielinga, ”Knowledge Engineering and Management - The CommonKADS Methodology”. MIT Press. (1999)
12. S. Staab, H.P. Schnurr, R. Studer and Y. Sure, ”Knowledge processes and ontologies”. IEEE Intelligent Systems 16 (1) (2001) 26-34.
13. L. Zhou, Q. E. Booker and D. Zhang, ”Toward Rapid Ontology Development for Underdeveloped Domains”. HICSS 2002: 106

14. C.W. Holsapple and K.D. Joshi, ”A collaborative approach to ontology design”. Commun. ACM 45 (2002) 42-47
15. P. Spyns, R. Meersman and M. Jarrar, ”Data modelling versus ontology engineering”. SIGMOD Record Special Issue 31(4), 12-17. (2002)
16. H. S. Pinto, S. Staab and C. Tempich, ”DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies”. ECAI 2004: 393-397.
17. K. Kotis and G. Vouros, ”Human-Centered Ontology Engineering: the HCOME Methodology”. International Journal of Knowledge and Information Systems (KAIS), 10(1): 109-131 (Published Online First: 9 Sept. 2005)
18. M. Missikoff and R. Navigli, ”Applying the unified process to large-scale ontology building”. In Proceed-ings of 16th IFAC World Congress (IFAC) (pp. 61-96). Amsterdam: Elsevier. (2005)
19. S. Karapiperis and D. Apostolou, ”Consensus building in collaborative ontology engineering processes”. Journal of Universal Knowledge Management, 1(3), 199-216 (2006).
20. A. Garcia Castro, P. Rocca-Serra, R. Stevens, C. Taylor, K. Nashar, M. Ragan and S. Sansone, ”The use of concept maps during knowledge elicitation in ontology development processes - the nutrigenomics use case”. BMC Bioinformatics, 7, 267-281. (2006)
21. Good, B., Tranfield, E.M., Tan, P.C., Shehata, M., Singhera, G., Gosselink, J., Okon, E.B., Wilkinson, M., ”Fast, cheap, and out of control: A zero curation model for ontology development”. In: Pacific Symposium on Biocomputing. (2006)
22. M. C. Su?rez-Figueroa, K. Dellschaft, E. Montiel-Ponsoda, B. Villaz?n-Terrazas, Z. Yufei, G. Aguado de Cea, A. Garc?a, M. Fern?ndez-L?pez, A. G?mez-P?rez, M. Espinoza, M. Sabou. NeOn Deliverable D5.4.1: ”NeOn Methodology for Building Contextualized Ontology Networks”. NeOn Project. Available online: http://www.neon-project.org. February 2008.
23. C. Tempich, H. S. Pinto and S. Staab, ”Ontology Engineering Revisited: An Iterative Case Study”. ESWC 2006: 110-124.