myOntology: The Marriage of Ontology Engineering and Collective Intelligence

Katharina Siorpaes and Martin Hepp

Digital Enterprise Research Institute (DERI), University of Innsbruck, Austria
katharina.siorpaes@deri.org, mhepp@computer.org

Abstract. Despite very active research on ontologies, only few useful ontologies can be found on the Web. The reasons for this are manifold, but a major obstacle is that ontology engineering environments impose high entrance barriers on users, and that the community does not have control over the ontology evolution. Wikis are a way to allow a wide range of users to contribute to Web representations without requiring more than basic Web editing skills. In the myOntology project, we propose the use of wiki technology in order to enable collaborative and community-driven ontology building by giving users with no or little expertise in ontology engineering the opportunity to contribute. In this paper, we describe the myOntology project in which the challenges of collaborative, community-driven, and wiki-based ontology engineering are investigated. Our approach combines the simplicity of wikis with intuitive visualization techniques and small yet efficient helper functionality plus consensus finding support exploiting the collective intelligence of a community.

Keywords: Ontologies, ontology engineering, collaborative ontology engineering, open ontologies, wikis

1 Introduction

Despite the active research on ontologies, only few useful ontologies can be found on the Web. The reasons for this are manifold including that ontology engineering environments impose high entrance barriers on users and the community does not have control over the ontology evolution. Wikis are a way to allow a wide range of users to contribute without requiring more than basic Web editing skills. In this paper we describe the myOntology project in which we use wiki technology in order to enable collaborative and community-driven ontology building by including users who have no or little expertise in ontology engineering. The remainder of this paper is organized as follows: in section 1, we outline the problem and motivate the paper. In section 1.3, we relate myOntology to previous works. In section 2, we describe the design principles and the architecture of our approach. In section 3, we sketch the implementation of the system. In section 4, we give a preliminary evaluation of our approach by comparing traditional ontology engineering to the myOntology vision. In section 5, we summarize our findings.

1.1 Motivation

Ontologies are widely regarded as the "backbone of the Semantic Web" [1], [2] and the intensity of research on ontologies and related topics is very substantial - which can easily be shown by searching for the terms "ontology" and "ontologies" on Google Scholar[1], yielding 370.000 respectively 133.000 scholarly documents or references. However, when searching the Web, only few mature, practically useful ontologies can be found. This phenomenon has been discussed e.g. in [3], in which four bottlenecks were identified: First, many relevant domains of discourse, such as in e-commerce, show a high degree of conceptual dynamics, i.e. it is hard to keep up with the pace of change in reality. Second, the costs and potential benefits of building and using ontologies may be unfairly distributed among actors. Third, a prerequisite for using an ontology and thus committing to its view of the world is to be able to understand the meaning of concepts and relations. This is problematic for many users, since they cannot easily figure out what they would be committing to when using a particular ontology file from the Web. Fourth, when reusing existing consensus, e.g. in the form of standards or encyclopedias, one faces intellectual property rights, which means that ontologizing such input will require legal agreements with the current owners.

Currently, the most popular approach towards ontology building is engineering-oriented: a small group of engineers carefully builds and maintains a representation of their view of the world. However, a community-oriented approach where multiple individuals work on an ontology collaboratively has several advantages over an isolated engineering-oriented approach:

1. A community can keep up with the pace of conceptual dynamics in a domain more easily. Users have an interest in keeping the ontology up-to-date and therefore have a strong motivation to contribute to the maintenance.
2. The burden of creating the ontology can be distributed more evenly across those benefiting from the ontology.
3. The user community is more likely to agree on a view of the world that is represented by the ontology. Therefore, it is likely that this community will also actually use and further develop the ontology as it is not a subjective conceptualization based on a outdated state of the world.

However, we are currently lacking tool support that is suitable for ontology construction by large groups of non-experts over the Web. On the other hand, the philosophy of wikis has been an enormous success for collaborative editing on a large scale, as the online encyclopedia Wikipedia[2] has shown. In the myOntology project, we propose to use the infrastructure and culture of wikis for a collaborative and open ontology building environment.

---

[1] http://scholar.google.com, retrieved on February 22, 2007
[2] http://wikipedia.org/

1.2 Our Contribution

In this paper, we (1) argue that the use of social software will very much improve the state of the art in ontology engineering. We then sketch (2) design principles as well as (3) major components of the myOntology platform. As the project is in an early state, we present a (4) preliminary roadmap for the implementation of the platform. Finally, we (5) evaluate our approach by comparing it with traditional ontology engineering and (6) introduce the notion of horizontal and vertical ontology engineering.

1.3 Related work

Our work is related to the following streams of research:

Collaborative ontology engineering: Substantial literature on collaborative ontology engineering already exists. However, the approaches we know of do not put the Wiki editing approach in the center of building ontologies collaboratively. [4], [5] describe Tadzebao and WebOnto: Tadzebao is a system that supports asynchronous and synchronous discussions on ontologies. While we agree that allowing dialogue is important, we question that users are willing to spend the time to agree on a concept definition by a non-structured discussion. In our opinion, the support for achieving consensus must be more subtle. WebOnto, a Java based ontology editor, complements Tadzebao by supporting collaborative browsing, creation, and editing of ontologies. [6] describe the DILIGENT knowledge process where ontology evolution and collaborative concept mapping are applied to deal with conceptual dynamics of domains. The ontology editor Protégé[3] is also available in a Web version. OntoEdit [7] is a collaborative ontology editing environment that allows multiple users to develop ontologies in three phases: kick-off, refinement, and evaluation/maintenance. It ensures consistency by blocking the part of the ontology that is being edited. [8], [9], and [10] propose an approach to community-driven ontology matching that allows different individuals to create mappings.

Semantic Wikis: [11] allows annotating links, typing of pages, and context dependent content adaptation. Additionally, it displays related pages. [12] allows annotating links, typing of pages, and context dependent content adaptation. Additionally, it displays related pages. [13] have the objective to make the knowledge within Wikipedia, the online encyclopedia, machine-accessible by adding semantic information, e.g. by typing of links. They propose the use of semantic templates in order to simplify annotation for users. Platypus Wiki [14] describe only the similarities between collaborative ontology engineering and wikis. The approach described in this paper differs clearly from all these approaches, as most of them aim at augmenting existing wiki content with semantics. The goal of our approach is to *use* wiki technology to collaboratively build ontologies.

[3] http://protege.stanford.edu/

2 The myOntology approach

The myOntology approach towards ontology engineering clearly differs from traditional, engineering-oriented approaches. In this section, we describe our approach. In section 2.1, we define some design principles which reflect the myOntology philosophy. In section 2.2, we summarize the major components of the project. In section 2.3, we describe how existing technology contributes to the project.

2.1 Design principles

The goal of the myOntology project is to establish the theoretical foundations of collaborative, community-driven ontology building using wikis. The following design principles constitute the philosophy of myOntology:

Community grounding: The engineering-oriented ontology building approach, where a small number of ontology engineers constructs the representation of the domain of discourse and releases the results at some point in time to a wider community of users has several disadvantages: ontologies representing domains comprising a high degree of conceptual dynamics need to be changed often. A centralized approach will be too slow to appropriately reflect these changes, since missing entries cannot be added to the ontology by any user who reveals the need for a new concept, but instead have to be added by a small group of ontology engineers. This will at all times hinder ontology evolution and produce outdated thus not usable ontologies.

Furthermore, different individuals might have different views of a domain and therefore conflicts arise. The engineering-oriented approach forces users to commit to the view of a small group of ontologists. Our goal is not only to allow co-existence and interoperability of conflicting views but more importantly support the community in achieving consensus similar to Wikipedia, where one can observe that the process of consensus finding is supported by functionality allowing discussion.

Another disadvantage of the traditional, engineering-oriented ontology building approach is the lack of communication between ontology creator and user who cannot easily grasp the intention of a concept. As visualized in Figure 1, usually the user only has the serialization of the given ontology, which at best contains a textual description of the intention of the contained concepts in the form of non-functional properties.

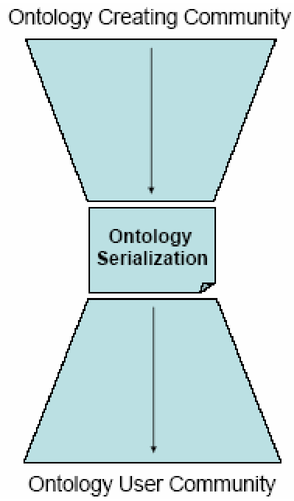

Fig. 1. The ontology perspicuity bottleneck [3]

A community based, de-centralized approach will reduce this problem, because ontology users can use the discussion that lead to the introduction of a concept as an additional hint to grasping the consensual meaning.

Ease of use: Existing traditional ontology engineering environments usually impose quite high entrance barriers on a user: a user with common Web-editing skills will not likely be able to create an ontology in e.g. Protégé[4] quickly. Social software offers the tools and paradigms in order to move from centralized towards de-centralized, community-grounded ontology building. Wikis allow many users to contribute easily with only basic Web-editing skills. However, the success of wikis also lies in many small but effective scripts that help the community build and maintain the corpus of knowledge, such as allowing discussion or the history function. We aim at developing small helper functionality that supports the community in developing the ontology.

Lightweight ontologies: The ontologies built with an open environment like myOntology might be rather simple models with a subsumption hierarchy. Though more expressive ontologies support more sophisticated reasoning we believe that also flat ontologies can be very useful. Even with a low degree of expressivity, such a framework would solve the problems described above: conceptual dynamics, cost vs. benefit, and perspicuity [3]. Furthermore, one has to keep in mind that a high expressivity allowing for complex language constructs might overstrain and scare most users away. When defining a suitable meta-model for a wiki-based ontology building framework, a trade-off between expressivity and usability needs to be made.

[4] http://protege.stanford.edu

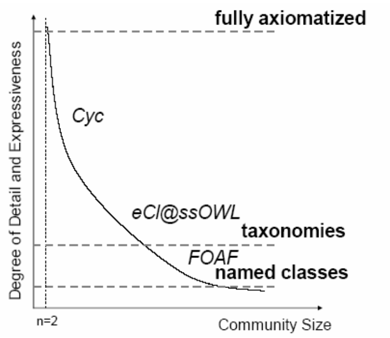

Fig. 2. The Expressivity-Community-Size Frontier [3]

[3] describes this as the expressivity-community-size frontier (Figure 2). It clearly shows that the more expressive an ontology is, the smaller is the user community as commitment costs are very high. Rather shallow and small ontologies such as FOAF have shown that ontologies have to comprise reasonable commitment costs.

2.2 Architecture

Addressing the problems delineated in the previous section involves divergent challenges, both within ontology engineering and beyond. In this section, we outline the major components of myOntology, which are visualized in Figure 3.

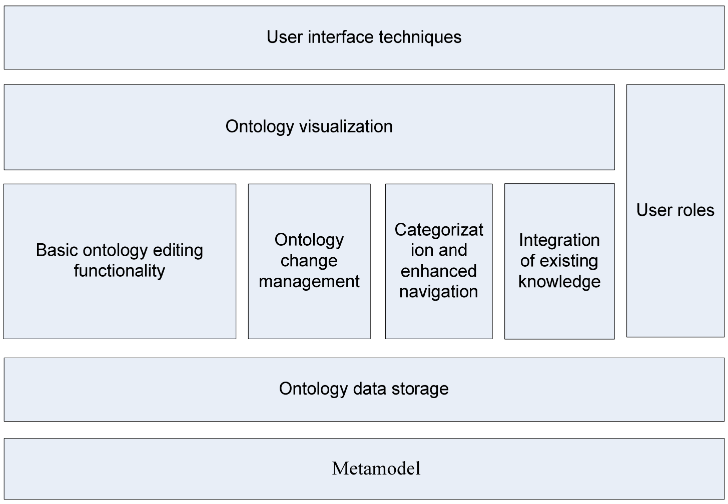

Fig. 3. myOntology components

Meta-model: We need to define an ontology meta-model that is suitable for a large audience. Obviously, non-expert users are not able to build highly axiomatized ontologies; as explained above flat ontologies can be useful as well. Additionally, reasoning support is desirable which comprises limitations concerning expressivity. The meta-model must support adding concepts, properties, and relations, as well as instances and several annotation properties. In order to support the upload of more expressive ontologies, elements that are not included in the meta-model will be preserved within annotation properties.

Ontology data storage: Ontology data as well as administrative data (e.g. user management) will be stored in a triple store using the myOntology ontology which represents the concepts and properties that are used within the environment. As myOntology will be open to the general public, the performance is especially important in order to preserve usability.

Basic ontology editing functionality: The focus especially in the first phase of the project lies on basic ontology editing functionality, such as adding and editing new classes and properties.

Ontology change management: Ontology change management comprises ontology evolution and versioning, as well as matching and mapping. In myOntology, we aim at community-supervised ontology change management: it is the community who track inconsistencies and remove them.

Categorization and enhanced navigation: In Wikipedia categories are used to enable better navigation and organization [17]. By very simple means, e.g. adding tags to definitions of ontology elements, a similar categorization system can be created in order to improve clarity as well as navigation additional to ontology browsing and concept search.

Integration of existing knowledge: In myOntology, we aim at integrating existing knowledge, such as references to Wikipedia articles. Furthermore, especially for e-commerce ontologies, the integration of eClassOWL [18] will allow much reuse of existing concepts. Dealing with homonyms and synonyms can be supported by using Wordnet [19]. Additionally, Google's mechanism for discovering spelling mistakes can add more value for the user.

Ontology visualization: In the collaborative ontology engineering paradigm it is extremely important that the meaning of a concept is obvious and easily understandable. In myOntology, ontology visualization techniques are emphasized additionally to a traditional, tabular view in order to help users understand the structure of an ontology, such as tag clouds and topic maps.

User roles: In myOntology, multiple kinds of roles which are necessary achieve consensus while editing and modifying an ontology are specified. We distinguish between four types of users: first, content consumers simply browse or use an ontology. Second, content providers regularly add new content. Third, content reviewers play an active role as well by reviewing existing content and participating in discussions. Fourth, super users are a few selected moderators, who act as administrators to the whole process and can, as a last resort, overrule the rest of the community. Mechanisms to assign user roles to users could be, e.g., an ontology modeling test, where users have to prove their abilities in ontology building.

User interface techniques: The importance of the design of the user interface is obvious as the audience of the project is very broad and non-technical. Most academic prototype implementations neglect the design their user interface. We will aim at building a nice and easy-to-use user interface based on existing work on interfaces, such as [20]. Additionally, we propose the use of multimedia elements. A natural language description of a concept supported by a picture conveys much more meaning than only text and improves disambiguity of informal concept definitions.

2.3 Contribution of existing technology components

MyOntology is an interdisciplinary project involving many research areas. We will make use of existing technology and state-of-the-art work in the most areas. For the myOntology meta-model a subset of OWL DLP [21] will be extended with some constructs from SKOS [22]. In order to support round tripping, more complex ontology elements that are not included in the meta-model are preserved by storing them using annotation properties. For storing ontology data, many different triple stores already exist. [23] present a detailed comparison of existing approaches. Substantial work has already been done in the area of traditional ontology development environments, such as Protégé. These tools provide excellent environments for skilled users allowing the creation of ontologies with a varying degree of expressivity. Existing ontology building tools will serve as a model for myOntology when it comes to basic ontology editing tasks, such as adding new classes, editing existing elements, etc. Handling ontology changes is probably the most complex challenge for myOntology. We will combine existing approaches with a community-supervised style of change management: similar to Wikipedia the community tracks inconsistencies and aligns concepts. Furthermore, we will exploit existing visualization techniques: implicit information contained in an ontology, such as the underlying structure of a data model or which instances are most closely connected is all contained in a graph. This information, though, is difficult, if not impossible, to extract from a text-based reading of the data. MyOntology will use techniques such as tag clouds and semantic networks.

3 Implementation

The myOntology project is in its early implementation phase. We follow the rapid prototyping paradigm due to the following reasons: (1) first results are visible immediately and work done can be verified instantly and (2) industrial partners can constantly check the development through an early quality assurance. First design decisions have been made: As a programming paradigm Java JSP will be used. PHP (like MediaWiki) was considered, however, Java gives more freedom when it comes to extensibility and the implementation of small helper scripts. Furthermore, Sesame is used as a triple store in order to store both, ontology data as well as administrative data. The first version of the prototype will be a wiki-based platform for browsing an ontology based on a minimal ontology meta-model. This meta-model supports classes (i.e., concepts / categories), attributes (i.e., slots for data type or object values assigned to classes), value categories as a special type of ontology classes, value instances of those value categories, and the “subclass Of” relation. The first prototype will allow users to suggest extensions or changes to any ontology element. Such change requests are pre-classified according to the context and semi-automatically processed by a privileged domain expert. In detail, users can recommend adding new classes, attributes, value types, as well as open feedback.

4 Evaluation

As the myOntology project is in an early stage and implementation work has only just begun, we evaluate our approach by comparing traditional ontology engineering to the myOntology approach. Additionally, we show the difference between *horizontal* and *vertical* ontology maintenance and why myOntology focuses on the support of horizontal ontology maintenance.

4.1 Traditional ontology engineering vs. the myOntology approach

The criteria for the evaluation are: (1) As shown by [24], many domains, especially in e-commerce, comprise a high degree of conceptual dynamics. *Timeliness* describes whether an ontology is up-to-date and hence useful. (2) *User participation* is an indicator how many individuals can contribute to the ontology evolution and especially if the control over the evolution is with the actual user community. (3) As ontologies are community contracts, the degree of *community grounding* depends on how agreed upon the representation of a domain is. (4) The *expressivity* of an ontology can range from flat collections of terms to abundantly axiomatized ontologies. (5) *Consistency:* ontology inconsistencies occur when the ontology is changed. Both design approaches comprise different risks for inconsistency.

Table 1. Traditional ontology engineering vs. the myOntology approach

	Traditional ontology engineering	**myOntology approach**
Timeliness	For an individual engineer or a group of engineers it is (a) more expensive and (b) more complex to keep the ontology up to date. Hence, ontologies maintained in a traditional ontology engineering approach, are more likely to be outdated.	A big community can keep up with the pace of conceptual dynamics more easily. This phenomenon can also be observed in Wikipedia. Therefore, myOntology will produce more up-to-date ontologies. This is a crucial feature in business domains.
User participation	In the engineering-oriented approach, only a small group of ontology engineers is involved. Users can only contribute by suggesting	In the myOntology approach the actual users of an ontology can contribute to and control the evolution of an ontology. This makes

	changes e.g. per e-mail or fax. This does not hinder ontology evolution but also consumes a lot of resources.	a commitment much easier for them.
Community grounding	Ontologies created in the traditional manner represent the view of the few ontologists working on the specification. Therefore, misconceptions are likely as well as a cleavage between the ontology and what the view of the community is.	Ontologies created with myOntology are real community contracts: like in Wikipedia, the community agrees on a specification supported by different functionality, such as discussion and history.
Expressivity	Depending on the skills of the engineers, in the traditional approach highly axiomatized, expressive ontologies can be created.	MyOntology will produce rather lightweight ontologies as most users can not be expected to be able to add axioms. However, this is not only a disadvantage: as shown by [24], a simpler ontology will have a bigger user community (which is desirable).
Consistency	In traditional ontology building, the resulting ontologies are more likely to be consistent as only a small group of skilled ontologist will work on the specification.	The more users the more likely inconsistencies occur. On the other hand, these inconsistencies can be tracked by the users themselves, like in Wikipedia.

4.2 Horizontal vs. vertical ontology maintenance

Regarding the expressivity of ontologies produced with myOntology, they will be rather lightweight. Too much expressivity will overstrain users and therefore hamper the creation of ontologies. In the following section, we describe the relation between horizontal and vertical ontology maintenance and the expressivity of ontologies and why myOntology can be described as a horizontal approach.

We distinguish between vertical and horizontal ontology maintenance (Figure 4): horizontal ontology maintenance can be understood as extending an ontology by concepts and properties but not in the level of detail or axiomatization. Vertical ontology maintenance emphasizes extending an ontology by axioms.

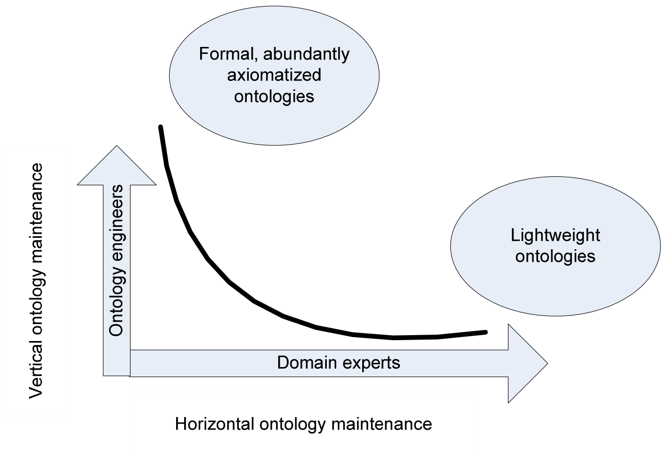

Fig. 4. Horizontal and vertical ontology maintenance

While in horizontal ontology maintenance, ontologies are rather shallow and lightweight, vertical ontology maintenance produces formal ontologies. Users will be able to create ontologies with a clear subsumption hierarchy with myOntology. In an expert mode it will be possible to add more complex constructs. However, the majority of ontologies created will be rather lightweight. The target groups for the project are the research community as well as domain experts with only basic Web editing skills, which makes myOntology a horizontal maintenance tool.

5 Conclusion

Ontologies are widely regarded as the backbone of the Semantic Web. However, only few ontologies can be found. Some reasons for this were outlined in the first section. The myOntology project described in this paper is supposed to enable more users to participate in creating and maintaining ontologies. Though these ontologies might not be highly axiomatized, they will be very useful to describe domains that can benefit from deploying ontologies, such as e-commerce. We introduce the notion of horizontal ontology maintenance opposed to vertical ontology maintenance, where myOntology is rather a horizontal approach. Open ontology engineering must have proper technical foundations but social and usability aspects must be considered as well. Wikis are social software that recently has been proven efficient and popular. Providing users with a usable tool that supports the community to establish community contracts on ontology definitions will result in more simple but useful ontologies that will be actually used in Web applications. The myOntology project aims at exploiting the collective intelligence of a community for ontology engineering.

Acknowledgments. This work has been funded by the Austrian BMVIT/FFG under the FIT-IT project myOntology (grant no. 812515/9284).

References

1. Fensel, D., Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. 2nd ed. 2004, Berlin etc.: Springer.
2. Ding, Y., et al., The Semantic Web - Yet Another Hip? Data & Knowledge Engineering, 2002. **41**(2-3 (June 2002)): p. 205-227.
3. Hepp, M., Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. IEEE Internet Computing, 2007. **11**(7): p. 96-102.
4. Domingue, J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. in 11th Knowledge Acquisition for Knowledge-Based Systems Workshop. 1998. Banff, Canada.
5. Aitchison, J., A. Gilchrist, and D. Bawden, Thesaurus construction and use: a practical manual. 4. ed. 2000, London: Aslib IMI.
6. Vrandecic, D., et al., The DILIGENT knowledge process. Journal of Knowledge Management, 2005. **9**(5): p. 85-96.
7. Sure, Y., et al. OntoEdit: Collaborative Ontology Engineering for the Semantic Web. in International Semantic Web Conference 2002 (ISWC 2002). 2002. Sardinia, Italia.
8. Gabel, T., Y. Sure, and J. Voelker, KAON - Ontology Management Infrastructure in SEKT Deliverable D3.1.1.a. 2004.
9. Zhdanova, A. and P. Shvaiko. User-driven Ontology Matching. in 3rd European Conference (ESWC2006). 2002. Madrid, Spain.
10. Zhdanova, A. and P. Shvaiko. Community-driven Ontology Matching. in Wiki meets Semantics workshop at the ESWC2006. 2006. Budva, Montenegro.
11. Dello, C., E. Paslaru Bontas Simperl, and R. Tolksdorf. Creating and using semantic content with Makna. in Wiki meets Semantics workshop at the ESWC2006. 2006. Budva, Montenegro.
12. Schaffert, S. Semantic Wikipedia. in 1st international workshop on World Wide Web (WWW2006). 2006. Manchester, UK.
13. Völkel, M., et al. Semantic Wikipedia. in 15th international conference on World Wide Web (WWW2006). 2006. Edinburgh, Scotland.
14. Bao, J. and V. Honavar. Platypus Wiki: a Semantic Wiki Wiki Web. in 3rd International Workshop on Evaluation of Ontology-based Tools (EON2004). 2004. Ancona, Italy.
15. Leuf, B. and W. Cunningham, The Wiki Way: Collaboration and Sharing on the Internet. 2001, London: Addison-Wesley.
16. Cunningham, W. Wiki Design Principles. [Web pages] 1995 May 3, 2007.
17. Wikipedia. Wikipedia Categorization. 2007; Available from: http://en.wikipedia.org/w/index.php?title=Wikipedia:Categorization&oldid=117808880.
18. Hepp, M. eCl@ssOWL. The Products and Services Ontology. 2006; Available from: http://www.heppnetz.de/eclassowl/.
19. Fellbaum, C., Wordnet: an electronic lexical database. 1998: The MIT Press.
20. Tidwell, J., Designing Interfaces. 2nd ed, ed. E. Odewahn and M. O'Brien. 2006: O'Reilly. 331.
21. Vrandecic, D., et al., DLP - An introduction, in Technical report. 2005, AIFB, Universität Karlsruhe: Karlsruhe.
22. W3C. Simple Knowledge Organisation System (SKOS). 2005; Available from: http://www.w3.org/2004/02/skos/.
23. Berners-Lee, T. URI References: Fragment Identifiers on URIs. 1997; Available from: http://www.w3.org/DesignIssues/Fragment.html.
24. Adida, B. and M. Birbeck. RDFa Primer 1.0. Embedding RDF in XHTML. W3C Working Draft 16 May 2006. 2006; Available from: http://www.w3.org/TR/xhtml-rdfa-primer/.