Community-Driven Ontology Management: DERI Case Study

Anna V. Zhdanova, Reto Krummenacher, Jan Henke, Dieter Fensel
DERI – Digital Enterprise Research Institute,
University of Innsbruck, Austria, and National University of Ireland at Galway, Ireland
{anna.zhdanova, reto.krummenacher, jan.henke, dieter.fensel}@deri.org

Abstract

We introduce the concept of community-driven ontology management and demonstrate the added value to conventional ontology management of being community-driven. Further, we present an implementation of an infrastructure supporting community-driven ontology management. The implemented infrastructure was deployed as a part of the intranet at DERI – Digital Enterprise Research Institute, and the community's response and behavior were observed. The results obtained prove feasibility and advantages of community-driven ontology management.

1. Introduction

Nowadays, numerous community Web portals related to business or leisure were created [13]. Many of them have proven to be highly popular and successful by acquiring millions of members. However, the existing portals are inflexible with respect to specification of user profiles, the content of the portals, organization and structuring of the content, and search options. The existing community Web portals declaratively specify what and how the users can contribute and find there. The specification given by the Web portal creators depends on their views of the domains, which are usually comprehensive, but are definitely restricted even in the best cases. Limitations in ontology management at the portal make it uninteresting for users and communities whenever they want to go beyond the portal creators' view of the domain.

A larger degree of flexibility and adaptability to the actual demands of portal members can be achieved by deploying Semantic Web technologies [3] and community-driven ontology management. At present, Semantic Web technologies are already applied to numerous community Semantic Web environments [5; 10; 11], which are by definition Semantic Web portals that are maintained by a community of users [14]. Meanwhile, as we show in this paper, current Semantic Web community environments can gain substantial benefits from integrating community-driven ontology management, i.e., enabling the community members to develop and maintain domain ontologies and to cross-link among different domains.

A community-driven ontology management infrastructure has to support developers and users in their efforts to create the Semantic Web content through constructing, populating and evolving the community environment. In practice, adding community-support to applications will naturally allow more users to create large amounts of semantic data and metadata, inherently extending the Semantic Web. One of the key motivations for community-driven ontology management is the fact that the Semantic Web is more likely to spread if large user communities are involved in its development. It is the goal of community-driven ontology management systems to provide the means and motivations to a large number of users to weave the Semantic Web, similar to the means and motivations available for the World Wide Web [2].

The paper is organized as follows. In Section 2, we describe our approach of community-driven ontology management. In Section 3, architecture and functionality of the implementation are presented. Our experience with practicing community-driven ontology management at DERI is described in Section 4. Section 5 concludes the paper.

2. Community-driven ontology management

In this section, the concept of community-driven ontology management is explained and the added value, with respect to conventional ontology management [6] is discussed.

The following four paragraphs describe the basic components of our approach. They are strongly related to conventional ontology management services, but enriched by acquisition and reuse of community-related information.

Community Ontology Editing service: This service is introduced for creating and maintaining ontologies and instance data. The front-end, a user-friendly interface, helps users to easily add and modify ontologies and its instances. The back-end consists of a storage system, such as databases, file systems or plain text files. A Community Ontology Editing service should enable consensual editing for multiple users and tight integration with semantic publishing and delivery components, allowing the involved parties to observe the ontology evolution. These requirements are grounded on a flexibility degree required to be granted to participants of community Semantic Web environments in order to freely evolve schemata and to influence community processes.

Community Ontology Storage and Query service: The goal of this component is to efficiently store and query small and large amounts of data and metadata by providing indexing, searching and query facilities for ontologies. Most of the currently available ontology storage and querying components are from a functional perspective similar to database or database management systems. In addition, Semantic Web search engines start to appear (such as Intellidimension Search Engine[1]). Enriching the existing search and query components with community-related information such as social networking would improve their performance and make them mature and more attractive to use.

As community members are generally not bound to a single community, they tend to publish personal and community-related data in a distributed way. The current focus in community storage and querying is thus maintaining distributed repositories with simple to use functionalities for aggregation, decomposition and discovery of information. A straightforward solution is hence to store and process the semantic (meta)data in distributed files (e.g., as FOAF[2] files on the Web).

Community Ontology Alignment service: A regular ontology aligner supports ontology mapping, which is however still mostly manual task. The front-end is a user interface for semi-automatic ontology mapping (such as recommendation lists and help for defining mapping rules). The back-end consists of an ontology inference engine. Annotated storage of previously acquired ontology mapping solutions and provision of an access to this storage for a wide reuse of acquired mappings are distinctive features of a Community Ontology Alignment service compared to a conventional ontology alignment service. Special ontologies are used to specify relevance, reusability and reliability of certain ontology mappings (employing social networking and statistical information). The ultimate goal of the alignment service is to enable knowledge services of external applications to gain benefit from the annotated mapping repositories and alignment sub-services and to ultimately cross-link communities.

Community Ontology Versioning service: A conventional versioning service represents different versions of ontologies, including backward consistency support and related instance versioning. The front-end provides a version information report, changes and their effects, e.g., the difference between two versions of the same ontology. The Community Ontology Versioning service needs to interoperate with the Community Ontology Editor, the Community Ontology Storage and the Community Query Manager as well as with pluggable inference engines for additional tasks such as consistency checking. On top of the ordinary functionality of a regular ontology versioning service, a community versioning service requires a set of simple understandable interfaces to be available and easily accessible through the Semantic Web. In addition, the service should track the changes taking place in distributed ontologies, report relevant inconsistencies and its resolutions or recommendations for resolutions.

Summarizing, we see the following values being added by community-driven ontology management compared to commonly used ontology management:

1) Ontology management is an expensive process. In community ontology management, the expenses are shifted from the portal maintainers to the communities employing ontologies. This shift results in ***adequate investment distribution*** among the ontology items (e.g., classes and properties). Specifically, the ontology items of higher importance to the community gain more support in terms of more associated resources, e.g., instance data, granularity in description, cross-ontology mappings.
2) The ontologies which are constructed, aligned and further operated by the communities ***represent the domain and connection with other domains more comprehensibly*** than the ontologies designed and maintained by an external knowledge engineer. External knowledge engineers are typically the bottleneck to the ontology comprehensiveness, as they are not capable to capture all the variety that might take place in a community and associated communities.
3) The community-driven ontology management approach provides a ***higher dynamicity and up-to-dateness*** to the outside-world changes in time, comparing to the conventional ontology management approach. When ontologies are constructed by external knowledge engineers, all the changes need to be captured and introduced by these engineers. With external knowledge experts, the delay in realizing and introducing the changes

[1] A Search Engine for the Semantic Web: http://semanticwebsearch.com

[2] FOAF project: http://foaf-project.org

might take days, weeks or even months. This delay is unacceptable for many dynamic domains, where new terminology regularly and rapidly changes (e.g., business or sport).

Thus, community-driven ontology management has major advantages over conventional ontology management. These advantages are gained by introducing an infrastructure that enables the communities to manage their ontologies. We have implemented a prototype for such an infrastructure and tested the implementation in a use case involving a community of co-workers in a research institute.

3. Implementation

The implemented infrastructure is designed as a component for a community Semantic Web portal providing ontology management facilities to the community members [15]. The infrastructure is built as a Java application including servlets and JSP technology, exploiting Jena 2 [4] for manipulating ontology schemata and instance data. Ontology Alignment API [7] with its built-in methods for automatic alignment of OWL ontologies was reused at the environment. An overview of the architecture of the community-driven ontology management infrastructure is shown in Figure 1.

The application requires involvement and population of domain-dependent and domain-independent ontologies, and service support for the portal's data and metadata (mostly, through publishing services for making the semantically enriched data human-readable).

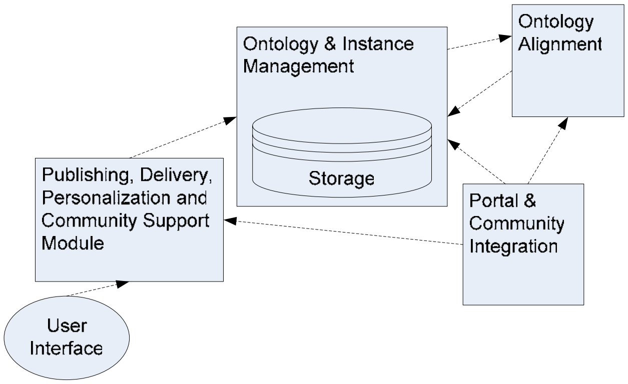

Ontology & Instance Management
Storage
Ontology Alignment
Publishing, Delivery, Personalization and Community Support Module
Portal & Community Integration
User Interface

Figure 1: Community-driven ontology management infrastructure

The infrastructure supports acquisition and exploitation of ontological structures by a community. In particular, the following community-driven ontology management functions are supported by our prototype:

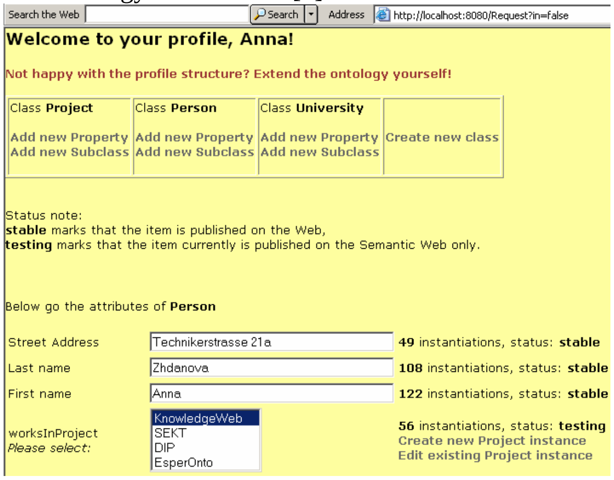
- *Editing* – the community members are enabled to extend the domain ontology via graphical Web interfaces, adding classes, subclasses, properties, instances and relating instances. The interfaces are generated dynamically depending on the structure and content of the RDFS or OWL domain and community-supporting ontologies. An example of an employed knowledge acquisition interface is shown in Figure 2.
- *Storage* – community related data are stored by means of a common centralized repository. Private instance data are stored and accessed in a distributed manner.
- *Alignment* – the implementation of the community alignment service[3] allows semi-automatic mapping between ontologies and saving the approved mappings in a publicly available storage, e.g., as an OWL file accessible over the Web. The community ontology alignment service provides a basis for interoperation across communities by linking ontology items used by various parties.
- *Versioning* – the instances are distinguished as community-related and individual-related. Community-related instances are generally relevant to more than one individual at a time and therefore are displayed to many individuals (e.g., information about research projects). Individual-related instances can be restricted to a particular individual (e.g., private phone numbers). Different versioning policies are applied to community-related and individual-related instances. For community instances after instance modification, a new instance is introduced and the visual name of the previous instance is modified. Private individual-related instances are rewritten when changed and a semantic change log is maintained at the community level.
- *Aggregation* – distributed content can be processed by the aggregation module to produce an input to other components, such as the publishing and information delivery component in Figure 1.

In addition, the following principles were taken into account in design and implementation of the community environment:

- *Ontology layer pyramid support* – Ontologies at different layers, such as user, community, portal layers, and different dimensions, such as profile and personalization dimensions, of the community environment are stored and evolved applying different storage and evolution policies [15].
- *Distributivity support* – A core principle of the environment design is to import/integrate ontologies and data from various locations on the Web. Therefore, ontology linking and metadata aggregation are supported.

[3] Ontology alignment service: http://align.deri.org

- *Automatic ontology population* – In order to reduce data input overheads for the members of the community Semantic Web environment, ontologies can be instantiated automatically. In particular, our experience with ontology instance acquisition from HTML Web pages employing the LixTo toolkit [1] indicates that acquisition of initial datasets is highly important for getting the users involved in community portal activities. At the same time, the means and sources of automatic information acquisition need to be chosen carefully, e.g., in our case the efforts spent on automatic acquisition of ontology instances for ca. 100 user profiles from existing HTML pages with LixTo were comparable with the efforts required for the manual insertion of the data.
- *Publishing and delivery of Semantic Web content* – The community members are enabled to introduce and see changes that were caused by their actions in the Semantic Web community environment also on the ordinary Web. The role of added-value publishing and delivery of information to the human user is crucial for user involvement in community-driven Semantic Web environments. Earlier practical experiments show that users do not get involved on a permanent basis with an environment that provides solely functionalities for collaborative ontology construction [8].



Welcome to your profile, Anna!

Not happy with the profile structure? Extend the ontology yourself!

Class **Project**	Class **Person**	Class **University**	
Add new Property Add new Subclass	Add new Property Add new Subclass	Add new Property Add new Subclass	Create new class

Status note:
stable marks that the item is published on the Web,
testing marks that the item currently is published on the Semantic Web only.

Below go the attributes of **Person**

Street Address	Technikerstrasse 21a	**49** instantiations, status: **stable**
Last name	Zhdanova	**108** instantiations, status: **stable**
First name	Anna	**122** instantiations, status: **stable**
worksInProject *Please select:*	KnowledgeWeb SEKT DIP EsperOnto	**56** instantiations, status: **testing** Create new Project instance Edit existing Project instance

Figure 2: Interface for knowledge acquisition in a community environment

- *Restricted access and user profiling* – Access policies for the community ontologies are normally required to protect the communities from unauthorized ontology management. At the simplest level, these policies can be supported by user profiling and password protection, as done in our implementation. Further, access policies can be implemented taking into consideration community and social networking information provided by the users [9].
- *Community-based consensus reaching* – By informing community members about the ontology evolution, the portal infrastructure facilitates the convergence of ontologies, i.e., it helps members to find a consensus in defining the shared ontologies.

4. DERI case study - experience

In this section, we describe the use case of a community-driven ontology management application, and our experiences and results from this case study are presented.

4.1. Case study description

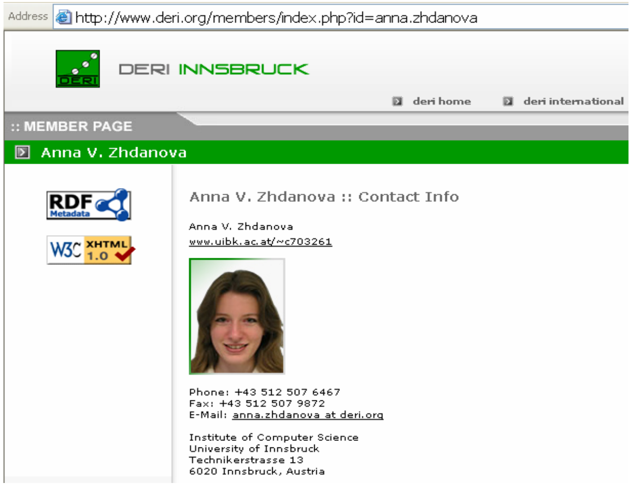
The case study was performed with the participation of the Digital Enterprise Research Institute (DERI)[4] employee community. DERI is a research organization with Semantic Web and Semantic Web service technology as major research areas. At the time the case study started (August 2004), DERI counted a community of 92 researchers and management team members: 35 members were affiliated with DERI Austria in Innsbruck and other 57 members with DERI Ireland in Galway. During the runtime of the case study (until January 2005), the quantity of DERI members exceeded one hundred employees.

Ontology management can only be driven by the community if fully integrated into the usage processes of a Semantic Web community environment. Provision of useful services for the community members is a must for an environment in order to be used and be community-driven For the DERI case study, provision of a personal, semantically-enriched homepage for each DERI member and a simple method to edit it via Web forms were settled as valuable services for the community. An advantage of choosing these services was the ease of their integration within the ontology management environment: employees were provided with the means to ontologically describe themselves, i.e. using a domain they know very well. Before the new environment was inaugurated, only 16 members of DERI Innsbruck and 13 members of DERI Galway had their own homepages. Thus, more than two thirds of the community's members did not have their personal information online. This was to a great extent due to the lack of Web design knowledge (e.g., among managers) or unwillingness to spend effort on personal homepage maintenance.

The infrastructure, as described in Section 3, was provided for usage to the DERI community. Initially, the

[4] DERI: http://www.deri.org

domain ontology of the DERI community environment contained one concept **Person**, with 10 properties, e.g., first and last names, phone numbers, etc. The concept **Person** was adapted from the core of the FOAF ontology, as FOAF is a highly popular ontology for describing people. At the beginning of the environment exploitation, the values for each DERI member are acquired automatically from the existing HTML pages with the LixTo toolkit [1] to reduce data input overhead for the community environment users. The community members are enabled to extend/create the community domain ontology by adding and reusing ontology classes, subclasses, properties, introducing and modifying community and individual instances, and relating available entities. The environment is branched to the DERI Web site, and by changing the data in the community environment, DERI members also change the data shown on their personalized Web site. An example of an automatically generated page is presented in Figure 3.



Anna V. Zhdanova :: Contact Info

Anna V. Zhdanova
www.uibk.ac.at/~c703261

Phone: +43 512 507 6467
Fax: +43 512 507 9872
E-Mail: anna.zhdanova at deri.org

Institute of Computer Science
University of Innsbruck
Technikerstrasse 13
6020 Innsbruck, Austria

Figure 3: Example of a personal Web site of a DERI member

4.2. Results

The DERI community environment is publicly available since October 20, 2004 (first release), and December 12, 2004, (the second version, improved on the basis of users' feedback). During a 70 days trial period (from October 20, 2004 to December, 31 2004) more than 60 feedback messages where expressed from users who tried and used the environment. The content and frequency of the requests are listed in Table 1.

Summarizing, most users expressed concerns regarding the human-readable Web part of the environment, and not the Semantic Web part. Users were requesting more information on the visual knowledge acquisition interfaces, asking to change visual characteristics of personal homepages, etc. Security and password protection was another issue addressed, as the first release of the environment was open to any Web user. With the second version, the problem was eliminated by securing all the data with a login and password system. Topics related to the Semantic or community-driven ontology management part of the environment (i.e., topics 5, 6, 7 in Table 1) were addressed less than the topics around human-readable Web. The few remarks directly addressing the community-driven ontology management were of two types: some users were confused and others troubled by the opportunity to influence a community environment more than usually. Confusions (i.e., when users do not know which effects can be achieved in the environment and how) can be significantly diminished by introducing better user interfaces naturally involving inexperienced users in the ontology management process. The users who are afraid of the potential effects of world wide community-driven ontology management (e.g., semantic spamming or other intentional, malicious misuse) should be provided with means to protect themselves against the undesired effects, e.g., by an opportunity to being subjected only to a limited number of communities and services.

Item	**Topics of requests**	**# of requests, version 1**	**# of requests, version 2**	**Total # of requests**
1	Ontology editor: functionality/visualization	7/1	0/3	11
2	Security, passwords	6	4	10
3	Human-readable Web-pages (looks, future pages)	7	2	9
4	Visibility of Semantic Web content on ordinary Web pages / content negotiation	2/6	0	8
5	URIs, community ontology	4	3	7
6	Considerations regarding usage of the environment	4	3	7
7	"Who is responsible for editing what?"	1	3	4
8	Bug reports	2	1	3
9	E-mail address protection from spam	0	2	2

Table 1: Classification of the requests to the DERI community environment

In Table 2, we summarize the actions of community members during the trial period, such as introduction of new ontology items and instances. Generally, we observed that basic entities like classes, subclasses, properties and instance data were successfully acquired and reused by the community. Examples of correctly acquired ontology items are listed in the second column of Table 2. However, certain propositions made by the community were not obvious to understand, and were considered as modeling mistakes. An example of "incorrectly" modeled ontology part is assigning names of specific projects (such as "SEKT" and "DIP") as property names for a class **Project**, which led also to "incorrect" instance assignment solutions as indicated in the third column of Table 2. However, "incorrect" modeling solutions were normally not supported (i.e., reused) by the community, which demonstrates the community's capability to replace an expert in selection of appropriate modeling solutions, advancing the community-driven ontology management approach.

Types of ontology items	**"Correctly" acquired**	**"Incorrectly" acquired**
Classes	Yes, e.g., "Hobby"	No
Subclasses	Yes, e.g., "Lecture" as a subclass of "Teaching"	Yes, e.g., "Manager" as a subclass of "Project"
Properties	Yes, e.g., "weblog" for a "Person"	Yes, e.g., property "SEKT" for "Project"
Instances	Yes (from initial ontology) – e.g., new instances for "Project", Yes (from community ontology) – e.g., "weblog" had more than 5 correct instantiations	No (from initial ontology), Yes (from community ontology), e.g., value "DIP" for property "SEKT" of a "Project"

Table 2: Community-driven ontology acquisition in the DERI case study

In the DERI case study, we have observed all the advantages of community-driven ontology management as discussed in Section 2:

1) *Adequate effort investment in ontologies* – The owners of the environment were free from constructing ontologies. All the ontology construction efforts were delegated to the community members. In comparison, conventional construction of an ontology for the same domain took more than half a year in time, involvement of several experts and multiple discussions [12], which brought unduly expenses to the working group without a guarantee of an adequate representation of the modeled community as a result.
2) *Comprehensive domain representation* – As ontology construction was delegated to the community members, only ontology items important for active community members were introduced and instantiated in the environment. The ontology which resulted from a community-driven ontology editing process was substantially different to the ontology constructed by the experts in the area [12]. The differences are obvious already at the upper ontology level: the expert-constructed ontology has **Agent**, **Event**, **Location**, **Publication**, **Tool** and **Topic** as the core classes, whereas the core classes of the community-constructed ontology are **Person**, **Project**, **Working Profile**, **Work**, **Teaching**, **Topic** and **Hobby**. These results indicate that experts are not capable to specify the community knowledge comprehensively, as a community would do it itself. For example, here, teaching activities were considered insufficiently important by the experts, and a wish of the research community members share information about their hobbies on the Semantic Web was unexpected and overlooked.
3) *Dynamicity and up-to-dateness* – In the case study, the ontology items and instances were introduced as soon as a community member missed an item or an instance and took an action to introduce it. Whereas with a typical, expert-controlled approach, setting of new items would take significant delay in appearance of the item as well as its extensions and instances. For example, in the case study, a property stating that a **Person** can have a weblog was introduced by a community member soon after the environment's launch, and shortly after this introduction, more than five weblog values were acquired from other community members.

5. Conclusion

We have introduced community-driven ontology management and its benefits. Specifically, we described an implementation supporting community-driven ontology management, and presented the results obtained from its application in a real-life scenario at a research institute (DERI). The results prove feasibility of community-driven ontology management systems and their maintenance. For the nearest future, we are interested in further development of the personalization, visualization and navigation functionalities. The received feedback demonstrates that improvement of knowledge acquisition interfaces has the potential to increase the infrastructure usability and to reduce the error rates in acquisition of ontologies and instance data. Moreover, applying community-driven ontology management to other use cases than an intranet environment is important for gaining further contributions and evolution processes from different communities. In our vision, for ensuring adequate support for ontologies, proper domain representation and dynamicity on the Semantic Web, the community-driven ontology management is crucial to be applied to community environments involving a large number of members and numerous heterogeneous communities, potentially up to arbitrary communities of arbitrary size from the whole Web.

Acknowledgements

The work is partly funded by the European Commission under the NoE KnowledgeWeb (IST-2004-507482) and IP DIP (IST-2004-507483). We thank our colleagues, especially Stefan Decker, Knud Möller, Andreas Harth, Sebastian Kruk, Daniel Bachlechner, Kathrin Prantner and Katharina Siorpaes, for fruitful discussions and inputs to this work. We are also thankful to all the DERI members who contributed to our project by using the community-driven ontology management infrastructure and/or by providing feedback.

References

[1] Baumgartner, R., Ceresna, M., Gottlob, G., Herzog, M., Zigo, V., 2003. „Web Information Acquisition with Lixto Suite”. In *Proceedings of the 19th International Conference on Data Engineering*, pp. 747-749.

[2] Berners-Lee, T., 1999. *Weaving the Web*, Harper, San Francisco.

[3] Berners-Lee, T., Hendler, J., Lassila, O., 2001. “The Semantic Web”. *Scientific American 284(5)*, pp. 34-43.

[4] Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K., 2004. “Jena: Implementing the Semantic Web Recommendations”, In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004),* Alternative track papers and posters, pp.74-83.

[5] Corcho, O., Gomez-Perez, A., Lopez-Cima, A., Lopez-Garcia, V., Suarez-Figueroa, M., 2003. “ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets”, In: Fensel, D. et al. (Eds.), *Proceedings of the Second International Semantic Web Conference*; Springer, LNCS 2870, pp. 802-817.

[6] Davies, J., Fensel, D., van Harmelen, F. (Eds.), 2002. *Towards the Semantic Web: Ontology-Driven Knowledge Management*, John Wiley & Sons.

[7] Euzenat, J., 2004. “An API for Ontology Alignment”, In *Proceedings of the Third International Semantic Web Conference*, Springer, LNCS 3298, pp. 698-712.

[8] Farquhar, A., Fikes, R., Rice, J., 1997. ”The Ontolingua Server: Tool for Collaborative Ontology Construction”, *International Journal of Human-Computer Studies, 46(6)*, pp. 707-728.

[9] Golbeck, J., Parsia, B., Hendler, J, 2003. “Trust Networks on the Semantic Web”, In *Proceedings of Cooperative Intelligent Agents 2003*, Helsinki, Finland.

[10] Hyvönen. E., Junnila, M., Kettula, S., Mäkelä, S., Saarela, S., Salminen, M., Syreeni, A., Valo, A., Viljanen, K., 2004. “MuseumFinland — Finnish Museums on the Semantic Web: User’s Perspective”, In *Proceedings of Museums and the Web 2004 (MW2004), Arlington*, Virginia, USA, March 31 – April 3, 2004.

[11] Maedche, A., Staab, S., Stojanovic, N., Studer, S., Sure, Y., 2003. “SEmantic portAL - The SEAL approach”, In: Fensel, D. et al. (Eds.), *Spinning the Semantic Web*, MIT Press, Cambridge, MA, pp. 317-359.

[12] Möller, K., Predoiu, L., Bachlechner, D., 2004. “Portal Ontology“, Semantic Web Portal Group deliverable 1, URL: http://sw-portal.deri.org/papers/deliverables/D1_v1.2PortalOntology.pdf.

[13] O'Murchu, I., Breslin, J.G., Decker, S., 2004. “Online Social and Business Networking Communities”, In *Proceedings of ECAI 2004 Workshop on Application of Semantic Web Technologies to Web Communities.*

[14] Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.-P., Studer, R., Sure, Y., 2000. "Semantic Community Web Portals”, *Computer Networks 33(1-6)*, pp. 473-491.

[15] Zhdanova, A.V., 2004. "The People's Portal: Ontology Management on Community Portals", In *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web (FOAF'2004),* 1-2 September 2004, Galway, Ireland, pp. 66-74.