CEUR Workshop Proceedings
http://ceur-ws.org
ISSN 1613-0073

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

Linked Open Ontology Services

Kim Viljanen, Jouni Tuominen, Mikko Salonoja and Eero Hyvönen

Semantic Computing Research Group (SeCo)
Aalto University, School of Science and Technology, and University of Helsinki
http://www.seco.tkk.fi/, firstname.lastname@tkk.fi

Abstract. Ontology repository systems are used for publishing and sharing ontologies and vocabularies for content indexing, information retrieval, content integration, and other purposes. However, interlinking these distributed repositories to provide global search and browsing over the repositories has not been made. In the spirit of Linked Open Data, we propose creating a network of Linked Open Ontology Services (LOOS) consisting of ontology repositories that publish their content using a shared API. To test the approach, we have defined an HTTP API and present a proof-of-concept implementation consisting of three client applications that are used for accessing a LOOS network of over 50 ontology servers, part of the Ontology Library Service ONKI.

1 Introduction

Ontology repositories have been considered a key resource for building a global infrastructure to enable the vision of the Semantic Web [1]. Many ontology repository systems exist for publishing and sharing ontologies and vocabularies for content indexing, information retrieval, content integration, and other purposes, e.g. Cupboard [2], BioPortal [3], OOR [4], and ONKI [5].

However, currently each ontology repository is a separate island, with no connections with other repositories. This means that, e.g. global search, browsing, or inference over the repositories can not be done, which creates a hindrance for using the ontologies globally. For example, searching for the concepts with the label "fish" from all existing ontology repositories around the Internet is currently not possible although many ontology repositories surely contain ontologies with matching concepts. Because the user may not find the correct ontology or concept for one's needs, it means that:

- the quality of annotations may decrease if the optimal concept is not found,
- redundant new ontologies are created if the existing ontologies are not found,
- interlinking of data decreases due to creating redundant ontologies, and
- merging data for semantic web applications becomes more difficult due to the need for ontology matching.

As a solution to the problem of how to access the repositories globally, in the spirit of the Linked Open Data[1] [6], we propose a Linked Open Ontology

[1] http://linkeddata.org

Services (LOOS) architecture consisting of ontology repositories that publish their content through common API, and thus making it possible to access the different ontology services in a distributed way.

In the following, we present the proposed architecture and API. Then, a proof of concept implementation of clients and servers is described. Finally, related work is discussed and contributions of the paper summarized.

2 LOOS API and Metadata About Services

The LOOS Network consists of ontology repositories that publish their content using a common, uniform LOOS API. A key idea of the LOOS API is to hide the ontology schema specific representations (such as OWL and RDFS) behind a uniform, simplified, SKOS-like representation of the key elements of the ontology: the concepts and their relations. Hiding ontological details makes it easier both to provide a uniform API to the ontologies and to display the search results from the underlying repositories in a uniform way to the user. After finding the matching ontology or concept, if the full power of a specific ontology representation is needed, the user can be directed to the specific ontology repository.

The main methods of the API are following:[2]

- *search*: search concepts
- *getLabels*: get the labels of a concept
- *getEquivalentConcepts*: get the equivalents of a concept
- *getConceptHierarchy*: get the concept hierarchy of a concept
- *getFullPresentation*: get all the information about a concept
- *getDirectory*: get the directory view of an ontology

The search method is used for finding concepts using various restrictions, such as text, concept type, or parent concept. The method returns a list of matching concepts. To get information about each concept, the properties and relations of a concept can be queried using methods such as *getLabels*. For efficiency, the *getFullPresentation* method returns all information about a certain concept in a single request. When browsing an ontology, an overview of the ontology is helpful. To support this, the *getDirectory* method returns an ontology-specific overview of the given ontology, consisting of e.g. the alphabetical ordering of the concept labels, manually defined grouping of concepts, or the top concepts of the ontology.

We propose implementing the LOOS API as a lightweight, stateless, and cacheable HTTP GET based API that returns data using the JSON format which provides an easy way to add LOOS support to ontology servers.

To find LOOS enabled services, metadata about the services is needed. We use the property *loos:APIBaseURL* for describing the URL of each LOOS endpoint. Additional information about the ontology such as title and description may be expressed using e.g. the Dublin Core metadata schema. Based on the metadata, e.g. a directory of LOOS services can be published.

[2] For the complete API documentation, check: http://www.yso.fi/loos/

3 The Proof-of-Concept Implementation

The ONKI SKOS ontology server has been used for publishing over 70 ontologies in the Finnish Ontology Library Service ONKI [7]. Global search to the ontologies was however not possible, because the ontologies were running as separate server instances. To solve this problem, and as a proof-of-concept implementation for the LOOS aproach, the LOOS API was implemented to the ONKI SKOS server and the server instances were described with LOOS metadata (see Fig. 1).

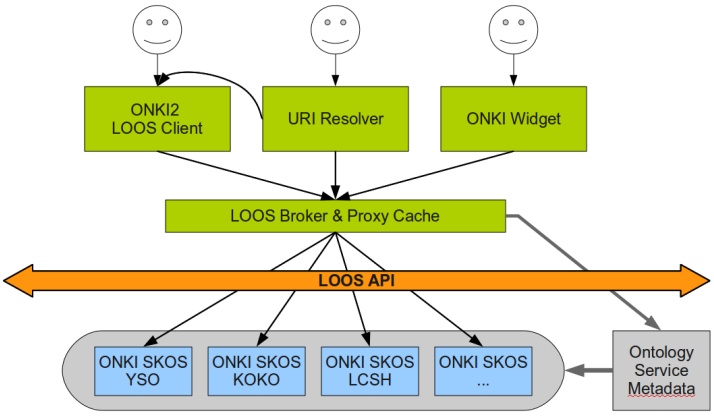

Figure 1. The proof-of-concept implementation of LOOS.

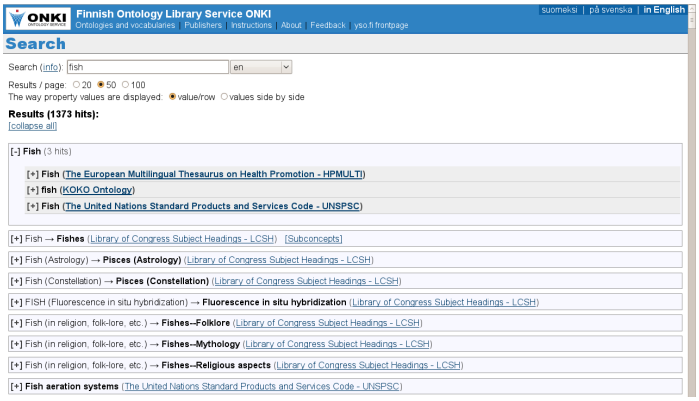

Fig. 2. The global LOOS search for "fish" matches many ontologies.

The ONKI2 Browser[3] is a global search and browsing user interface for accessing the LOOS network of ontology servers (the ONKI SKOS servers) in a uniform way . For example, making a global query to all ontology servers can be done (see Fig. 2). ONKI2 was mostly implemented using PHP[4].

Another client is the JavaScript-based ONKI Selector widget [8] for adding ontological concept search to HTML forms which now supports using the LOOS network as a back-end. As a third client, we implemented also a URI resolver for dereferencing the end-user's ontology concept URI requests to a suitable representation provided via the LOOS network, such as HTML or RDF.

To make client implementation easier, a broker for accessing the LOOS network was implemented. It provides a registry of LOOS enabled ontology servers based on the ontology service metadata, a single access point for using the LOOS network and a cache[5] to speed up potentially slow HTTP requests to individual ontology servers. Based on the registry, the broker directs LOOS requests to relevant back-end ontology servers.

4 Discussion

Compared to more general methods of accessing RDF data, such as SPARQL[6] and Linked Data [6], the LOOS approach focuses on ontologies. For example, when querying for the concept hierarchy of a concept via the LOOS API, one does not need to know what RDF properties (e.g. *rdfs:subClassOf* or *skos:broader*) are used in the data to express the hierarchical relations between concepts. The LOOS API restricts the set of possible queries, which makes it easier to implement and use the API in the ontology servers and the client applications.

APIs for accessing ontologies and vocabularies presented previously include the SKOS API[7] and the OWL API[8]. Compared to them, the LOOS API provides a higher abstraction, independent from specific ontology languages. Compared to the API's of BioPortal [3], Swoogle[9] and Watson[10], the goal of LOOS is to create a network of ontology servers based on a shared API that is implemented by all services. Therefore, the LOOS API focuses on a few basic methods that reflects the basic functionality of ontology repositories, e.g. concept search. In future, the LOOS API should be made compatible with repository specific APIs.

Ontology servers such as BioPortal and Cupboard support publishing interlinked ontologies, but the ontologies have to be uploaded into a centralized service for a global search. In contrast, in the LOOS approach ontologies can be published using a ontology service that is optimized for the specific ontology and

[3] http://www.yso.fi/onki2/?l=en
[4] http://www.php.net/
[5] As a proxy cache we used Varnish: http://varnish-cache.org/
[6] http://www.w3.org/TR/rdf-sparql-query/
[7] http://www.w3.org/2001/sw/Europe/reports/thes/skosapi.html
[8] http://owlapi.sourceforge.net/
[9] http://swoogle.umbc.edu/
[10] http://watson.kmi.open.ac.uk/

the user's needs while publishing the ontology service's basic functionality via the LOOS API to connect the ontology service to a global network of ontology services.

The loosely coupled LOOS architecture has turned out to be a flexible solution which makes it easy to implement additional clients when needed. Making multiple HTTP requests to back-end servers may be slow, but in our test implementation this lag has not been a problem thanks to the proxy cache between the clients and the back-end servers.

To conclude, this paper argues that the various ontology servers on the web should be made accessible using a common API that would provide a simple but universal methods for accessing the ontology content. As a solution, we propose the LOOS API and a metadata schema for describing the services.

Acknowledgements This work is part of the National Semantic Web Ontology project in Finland[11] (FinnONTO, 2003-2012), funded mainly by the National Technology and Innovation Agency (Tekes) and a consortium of 38 organizations.

References

1. Hyvönen, E., Viljanen, K., Tuominen, J., Seppälä, K.: Building a national semantic web ontology and ontology service infrastructure—the FinnONTO approach. In: Proceedings of the ESWC 2008, Tenerife, Spain, Springer–Verlag (2008)
2. d'Aquin, M., Lewen, H.: Cupboard - a place to expose your ontologies to applications and the community. In: Proceedings of the ESWC 2009, Heraklion, Greece, Springer–Verlag (June 2009) 913–918
3. Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., Musen, M.A.: BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research **37**(Web Server issue) (2009) 170–173
4. Baclawski, K., Schneider, T.: The open ontology repository initiative: Requirements and research challenges. In: Proceedings of Workshop on Collaborative Construction, Management and Linking of Structured Knowledge at the ISWC 2009, Washington DC., USA (October 2009)
5. Viljanen, K., Tuominen, J., Hyvönen, E.: Ontology libraries for production use: The Finnish ontology library service ONKI. In: Proceedings of the ESWC 2009, Heraklion, Greece, Springer–Verlag (2009)
6. Bizer, C., Cyganiak, R., Heath, T.: How to publish linked data on the web. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/ (July 27 2007)
7. Tuominen, J., Frosterus, M., Viljanen, K., Hyvönen, E.: ONKI SKOS server for publishing and utilizing SKOS vocabularies and ontologies as services. In: Proceedings of the ESWC 2009, Heraklion, Greece, Springer–Verlag (2009)
8. Viljanen, K., Tuominen, J., Hyvönen, E.: Publishing and using ontologies as mashup services. In: Proceedings of the 4th Workshop on Scripting for the Semantic Web (SFSW2008), 5th European Semantic Web Conference 2008 (ESWC 2008). (June 1-5 2008)

[11] http://www.seco.tkk.fi/projects/finnonto/