Oyster - Sharing and Re-using Ontologies in a Peer-to-Peer Community

Raúl Palma[1], Peter Haase[2]

[1] Ontology Engineering Group, Laboratorio de Inteligencia Artificial. Facultad de Informática.
Universidad Politécnica de Madrid. Spain.
[2] Institute AIFB, University of Karlsruhe, Germany.

Abstract. This paper presents Oyster, a Peer-to-Peer system for exchanging ontology metadata among communities in the Semantic Web. We describe how Oyster assists researchers in re-using existing ontologies, and how Oyster exploits semantic web techniques in data representation, query formulation, query result presentation to provide an online solution to share ontologies.

1 Introduction

Currently, ontology re-use is rather difficult, as it is hard to find and share ontologies available among the community. This leads to the problem of having many isolated ontologies created by many different parties. Besides the costs of the duplicate efforts this also hampers interoperability between ontology-based applications. Oyster[1] is a Peer-to-Peer application that exploits semantic web techniques in order to provide a solution for exchanging and re-using ontologies. To achieve this, Oyster implements a proposal for a metadata standard, so called Ontology Metadata Vocabulary (OMV)[2] [4] which is based on discussions and agreement in the EU IST thematic network of excellence Knowledge Web[3] as the way to describe ontologies. Exchanging ontology metadata is an interesting use case for a Peer-to-Peer application on the Semantic Web application for the following reasons: The information sources (ontologies) are geographically distributed among the community, and developers are willing to share the information about the ontologies they created provided they do not have to invest much work in doing so, while at the same time they are able to mantain the ownership of their ontologies. As a Peer-to-Peer system, Oyster further benefits from the following characteristics: no need for a centralized server (thus avoiding a bottleneck for both computational performance and information update), robustness against failure of any single component, and scalability both in data volumes and the number of connected parties.

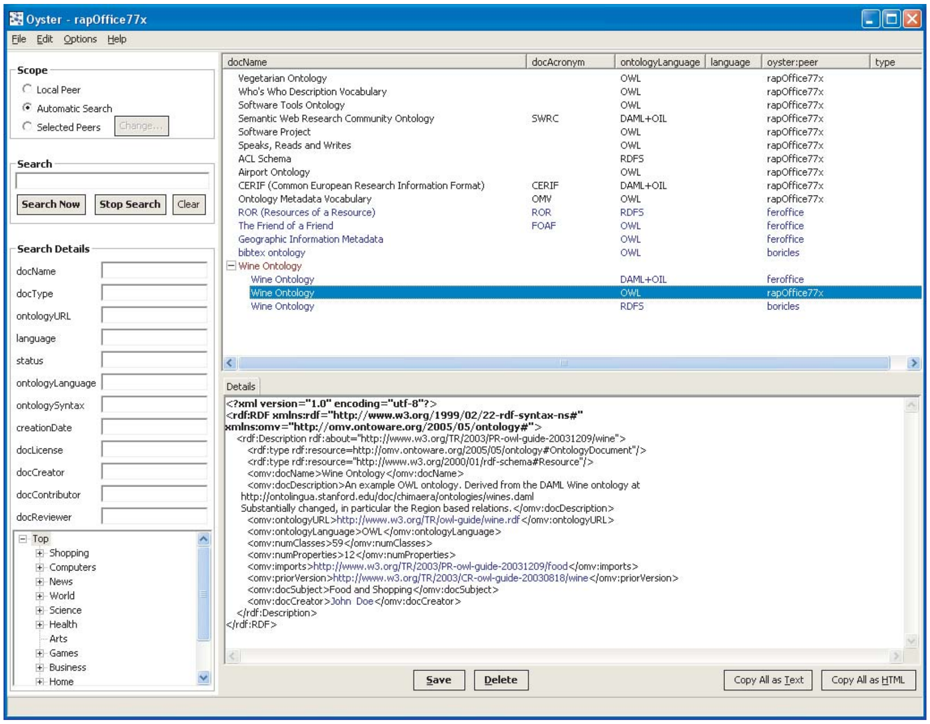
Finally, since ontologies can be represented in different languages (such as OWL[5], DAML+OIL[1], RDF-S[2]), Oyster provides the possibility to exchange heterogeneous information through the use of the metadata standard.

[1] Oyster is freely available for download under `http://oyster.ontoware.org/`
[2] The OMV ontology is available at `http://ontoware.org/projects/omv`
[3] `http://knowledgeweb.semanticweb.org/`

2 Oyster

Oyster provides an innovative solution for sharing and re-using knowledge (i.e. ontologies) which is a crucial step to enable Semantic Web.

The Oyster system has been implemented as an instance of the Swapster system architecture[4]. In Oyster, ontologies are used extensively in order to provide its main functions (importing data, formulating queries, routing queries and processing answers).



docName	docAcronym	ontologyLanguage	language	oyster:peer	type
Vegetarian Ontology		OWL		rapOffice77x	
Who's Who Description Vocabulary		OWL		rapOffice77x	
Software Tools Ontology		OWL		rapOffice77x	
Semantic Web Research Community Ontology	SWRC	DAML+OIL		rapOffice77x	
Software Project		OWL		rapOffice77x	
Speaks, Reads and Writes		OWL		rapOffice77x	
ACL Schema		RDFS		rapOffice77x	
Airport Ontology		OWL		rapOffice77x	
CERIF (Common European Research Information Format)	CERIF	DAML+OIL		rapOffice77x	
Ontology Metadata Vocabulary	OMV	OWL		rapOffice77x	
ROR (Resources of a Resource)	ROR	RDFS		feroffice	
The Friend of a Friend	FOAF	OWL		feroffice	
Geographic Information Metadata		OWL		feroffice	
bibtex ontology		OWL		boricles	
Wine Ontology					
Wine Ontology		DAML+OIL		feroffice	
Wine Ontology		OWL		rapOffice77x	
Wine Ontology		RDFS		boricles	

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:omv="http://omv.ontoware.org/2005/05/ontology#">
  <rdf:Description rdf:about="http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine">
    <rdf:type rdf:resource=http://omv.ontoware.org/2005/05/ontology#OntologyDocument"/>
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
    <omv:docName>Wine Ontology</omv:docName>
    <omv:docDescription>An example OWL ontology. Derived from the DAML Wine ontology at
   http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml
   Substantially changed, in particular the Region based relations.</omv:docDescription>
    <omv:ontologyURL>http://www.w3.org/TR/owl-guide/wine.rdf</omv:ontologyURL>
    <omv:ontologyLanguage>OWL</omv:ontologyLanguage>
    <omv:numClasses>59</omv:numClasses>
    <omv:numProperties>12</omv:numProperties>
    <omv:imports>http://www.w3.org/TR/2003/PR-owl-guide-20031209/food</omv:imports>
    <omv:priorVersion>http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine</omv:priorVersion>
    <omv:docSubject>Food and Shopping</omv:docSubject>
    <omv:docCreator>John  Doe</omv:docCreator>
  </rdf:Description>
</rdf:RDF>
```

Fig. 1. Oyster screenshot

Creating and importing metadata: Oyster enables users to create metadata about ontologies manually, as well as to import ontology files and to automatically extract the ontology metadata available, letting the user to fill in missing values. For the automatic extraction, Oyster supports the OWL, DAML+OIL and RDF-S ontology languages. The ontology metadata entries are aligned and formally represented according to two ontologies: (1) the proposal for a metadata standard OMV which describes the properties of the ontology, (2) a topic hierarchy (i.e. DMOZ topic hierarchy), which describes specific categories of subjects to define the domain of the ontology.

[4] http://swap.semanticweb.org/

Formulating queries: As shown in the left pane of the screenshot, the user can search for ontologies using simple keyword searches, or using more advanced, semantic searches. Here, queries are formulated in terms of these two ontologies. This means queries can refer to fields like name, acronym, ontology language, etc. (using the ontology document metadata ontology) or queries may refer to specific topic terms (using the topic hierarchy i.e. DMOZ).

Routing queries: As shown in the upper left pane of the screenshot, the user may query a single specific peer (e.g. their own computer, because they can have many ontologies stored locally and finding the right one for a specific task can be time consuming, or users may want to query another peer in particular because this peer is a known big provider of information), or a specific set of peers (e.g. all the member of a specific organization), or the entire network of peers (e.g. when the user has no idea where to search), in which case queries are routed automatically in the network. In the latter case, queries are routed through the network depending on the expertise of the peers, describing which topic of the topic hierarchy (i.e. DMOZ) a peer is knowledgeable about. In order to achieve this expertise based routing, a matching function determines how closely the semantic content of a query matches the expertise of a peer [3].

Processing results: Finally, the results matching query are presented in a result list (c.f. upper right pane in the screenshot). The answer of a query might be very large, and contain many duplicates due to the distributed nature and potentially large size of the Peer to Peer network. Such duplicates might not be exactly copies because the semi structured nature of the metadata, so the ontologies are used again to measure the semantic similarity between different answers and to remove apparent duplicates. Then a merged representation that combines the knowledge from the individual and potentially incomplete items is presented to the user. The details of particular results are shown in the lower right of the screenshot. The user can integrate results of a query into their local repository for future use. This information may in turn be used later to answer queries by other peers. Also, as proposed by OMV, all the specific realizations of an ontology can be grouped by the same base ontology to organize the answer.

3 Ontology Metadata Vocabulary in Oyster

Oyster applies semantic web technologies in order to build an online application for exchanging information that will assist users in building applications faster. In particular, it targets to re-using existing ontologies. In order to achieve this objective, Oyster provides an infrastructure for storing, sharing and finding ontologies making use of the proposal for a metadata standard OMV.

OMV distinguishes between an ontology base and an ontology document. This separation is based on the observation that any existing ontology document has some kind of core idea (conceptualisation) behind. From an ontology engineering perspective, initially a person develops such core idea of what should be modeled (and maybe how) in his mind. Further, this initial conceptualisation might be discussed with other persons and after all, an ontology will be realised using an ontology editor and stored in a specific format. Over time, there might be created several realisations of this initial conceptualisation in many different formats, e.g. in RDF-S[2] or OWL[5]. Therefore, an Ontology Base (OB) represents the abstract or core idea of an ontology, so called conceptualisation, and it describes the core properties of an ontology, independent from any implementation details. While an Ontology Document (OD) represents a specific realization of an ontology base, describing properties of an ontology that are related to the realization or implementation.

The distinction between an OB and OD leads to an efficient mechanism, e.g. for tracking several versions and evolvements of ontologies as well as for different representations of one knowledge model (conceptualisation) in different languages.

OMV also models additional classes required to represent and support the reuse of ontologies by such metadata vocabulary, especially in the context of the Semantic Web. Hence, OMV further models classes and properties representing environmental information and relations such as Person, Organisation, Party, OntologyEngineeringTool, OntologySyntax, OntologyLanguage, OntologyType. For a description of the complete OMV ontology we refer the reader to [4].

4 Conclusion

Sharing an re-using ontologies within communities is a critical task, which previously was rather difficult because of the heterogeneity, distribution and diverse ownership of the ontologies as well as the lack of sufficient metadata. In this paper, we have summarized the implementation of Oyster, a semantics-based Peer-to-Peer system for the exchange of ontology metadata, that exactly addresses these challenges. Oyster exploits semantic web technologies to provide a solution for re-using ontologies. It builds on a proposed standard for metadata for describing ontologies. Oyster is already being applied in the Knowledge Web project with partner across the european union.

For more information about Oyster, we refer the reader to http://oyster.ontoware.org/.

Acknowledgments Research reported in this paper has been partially financed by the Knowledge Web project FP6-507482. We would like to thank our colleagues for fruitful discussions.

References

1. DAML+OIL (March 2001) reference description, 2001. W3C Note, available at http://www.w3.org/TR/daml+oil-reference.
2. D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Rec. 10 February 2004, 2004. available at http://www.w3.org/TR/rdf-schema/.
3. P. Haase, R. Siebes, and F. van Harmelen. Peer selection in peer-to-peer networks with semantic topologies. In *Proceedings of the International Conference on Semantics in a Networked World (ICNSW'04)*, Paris, June 2004.
4. J. Hartmann, R. Palma, Y. Sure, M. Suarez-Figueroa, P. Haase, A. Gomez-Perez, and R. Studer. Ontology metadata vocabulary and applications. In *Proc. of the Workshop on Web Semantics (SWWS'05), First IFIP WG 2.12 and WG 12.4 Agia Napa, Cyprus*, 2005.
5. M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Rec. 10 February 2004, available at http://www.w3.org/TR/owl-guide/.