A Web-based Platform for Collaborative Ontology Management

Sören Auer
University of Leipzig,
Leipzig, Germany
auer@informatik.uni-leipzig.de

Abstract

With the Ontology Web Language there now exists a semantic web language for knowledge representation on the Web. Although there are some application programming interfaces and tools for OWL available, a framework for parsing, storing, querying, manipulating, serving and serializing OWL knowledge bases in a collaborative web enabled environment for the most distributed web application platform (Linux/Apache/ PHP/MySQL) was still missing. We give an overview on pOWL, a web-based semantic web development platform for PHP, which may be used as a foundation framework for semantic web applications.

1 Introduction

The broad application of ontologies as shared terminological knowledge representations is one of the main strategies of the semantic web paradigm. With OWL [2] there exists now a W3C standard for defining web enabled ontologies which fits in the semantic layering of web languages [3]. Although there are some OWL-editors and APIs available, most of them are complicated to deploy or handle, don't support strategies for collaborative, distributed development of ontologies, are not Open Source. Since PHP is the most distributed web development technology, the semantic web paradigm will probably only be successful in a broad perspective if there are tools available tightly interacting with this language. The goal of this poster is to present a software framework named pOWL [1] which meets this requirement and aims to deliver an easy-to-deploy and easy-to-use, web-based ontology management solution to the scientific and Open Source community.

2 Architecture

pOWLs architecture is n-tier, while trying to minimize dependencies and supplying clean interfaces between tiers. It consists of the following tiers:

- *pOWL store* – SQL compatible relational database backend for storing model metadata and RDF triples
- *RDFAPI [4], RDFSAPI, OWLAPI* – layered APIs for handling RDF, RDF-Schema and OWL
- *pOWL API* – containing classes and functions to build web applications on top of those APIs
- *User interface* – a set of PHP pages combining widgets for accessing model data in a pOWL store.

3 Model Evolution and Versioning

To enable domain experts to collaboratively develop shared conceptualizations a key requirement is to support a versioning strategy. According to pOWL every editing action can be decomposed into smaller editing actions and finally into adds and removes of RDF triples to or from the RDF model. The following example illustrates this:

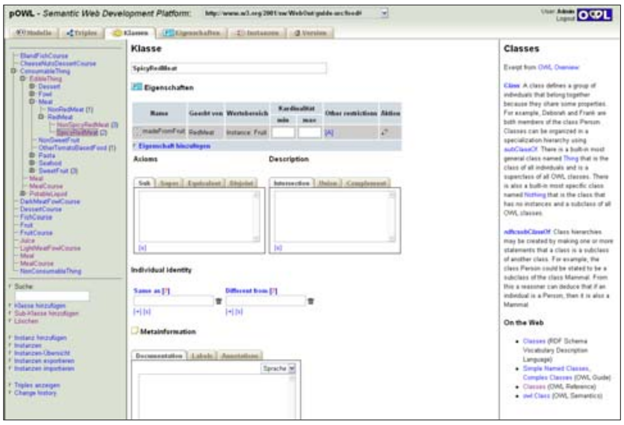
```
Class "owl:wine" updated
 Labels added
  Label for language "de" added: "Wein"
   Statement added:
    <owl:wine> <rdfs:label> "Wein"
 Annotation property
  "rdfs:seeAlso" changed
   Statement removed:
    <owl:wine> <rdfs:seeAlso>
                    "wn-concept:103880346"
   Statement added:
    <owl:wine> <rdfs:seeAlso>
        "http://en.wikipedia.org/wiki/Wine"
```

pOWL enables rollback of every particular editing action by determining if the involved triples are still present (if added) or still missing (if removed). For linear undo-redo operations this strategy is sufficient, even despite pOWL prevents triples to be added twice. When used selectively this undo-strategy has currently to be employed with more care.

4 Customizability

Instead of using configuration files or special database tables for customization, user authorization and preference management as well as user interface translation purposes and module integration, pOWL uses a "system ontology" for storing such knowledge. Three examples for knowledge in this system model are given:

- Instances of the class "Label" contain translations for all texts presented on the user interface.



- Instances of classes “User” and “Group” are responsible for authorization and storing of preferences.
- Further classes are used to store configuration data of pOWL modules and widget plugins.

pOWL may be extended by software modules which are placed in a subfolder of the pOWL distribution. An OWL based semantic web content management prototype is currently implemented exemplarily as a pOWL module.

5 User Interface and Widgets

The user interface is arranged in tabs, each tab representing a different view on the knowledge base. In pOWL tabs for viewing and editing models, RDF triples, classes, properties, instances and versioning information are included (additional ones may be added in a modular manner). pOWL provides a comprehensive library of plug-ins for comfortable editing of data: single and multiple line text editing, radio, checkbox and drop-down widgets, WYSIWIG HTML editing and selection of arbitrary resources.

RDQL [5] is an implementation of an SQL-like query language for RDF. pOWL contains a RDQL query builder to question the knowledge base. It constraints the users input to values making sense in the currently selected knowledge base and which make up a syntactically correct RDQL query.

6 Different Points of View on a Model

From the programmer’s as well as from the knowledge engineer’s or user’s point of view an OWL knowledge base may be seen through quite different glasses:

- *Triples View* – Triples or RDF statements are related to natural language sentences consisting of subject, predicate and object. In RDFS and OWL such statements are used to define higher level objects and to establish relations between them.
- *Database View* – An OWL knowledge base may be seen like an object-relational database, where OWL classes correspond to tables, properties to their columns and instances to rows in these tables.
- *Description Logic Axioms View* – OWL DL is a restriction of RDF to only such triples which have a representation as Description Logic axioms. pOWL detects such triples and represents them to the user using the DL terminology. However pOWL currently does not constrain users to a specific RDF subset (such as RDFS, OWL Lite/DL/Full).

The pOWL user interface provides special widgets supporting the user in viewing and editing information according to these different viewpoints. These widgets are bound to corresponding RDFSAPI and OWLAPI calls providing this functionality on the API level.

7 Scalability

pOWL is designed to work with ontologies of arbitrary size (only limited by disk space). This requires that only those parts of the ontology are loaded into main memory, which are required to render the information requested by the user as a web page. The following benchmarks were taken on a recent system with Windows, Apache, PHP and MySQL software installed.

Model	Triples	Import	Classes	Class-Tree Rendering
Wordnet[1]	473 528	624 s	6	0.30 s
NCI Cancer[2]	463 878	597 s	27 652	0.46 s
UNSPSC[3]	82 500	82 s	16 499	1.06 s

8 Conclusion

For domain experts and knowledge engineers pOWL provides an easy deployable and easy to use, web-based ontology editing and publishing solution. pOWL supports collaborative work with ontologies as well as observing the ontology evolution. pOWL is scalable and can be used even with extremely large knowledge bases. It may be customized according to specific needs. For PHP developers pOWL offers a comprehensive framework of functionalities for parsing, storing, querying, manipulating, serving and serializing RDFS and OWL ontologies. It may function as a basis for more domain specific web-based ontology applications. pOWL has been downloaded over 400 times since February 2004 (development start). It is available under GNU Public Licence (GPL). For future releases it is planned to integrate inferencing capabilities and to apply pOWL for OWL-S and web service development.

References

1. Auer, S.: pOWL Homepage, The Web, 2004, http://powl.sourceforge.net/
2. Bechhofer, S. et al.: Web Ontology Language (OWL) Reference version 1.0. W3C (2004), http://w3.org/TR/owl-ref/.
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 284 (2001) 35–43
4. Oldakowski, R., Bizer, C.: RAP: RDF API for PHP, To be published at “Workshop on Interpreted Languages”, 2004.
5. Seaborne, A.: RDQL - A Query Language for RDF, W3C Member Subm., 2004. http://w3.org/Submission/RDQL/

[1] http://www.semanticweb.org/library/wordnet/
[2] http://www.mindswap.org/2003/CancerOntology/
[3] http://www.cs.vu.nl/~mcaklein/unspsc/unspsc/