Creation of a browsable prototype of the portal

Zheng Jijuan, Robert Meersman
StarLAB, Vrije Universiteit Brussels, Brussels
{jzheng,meersman}@vub.ac.be

Rudi Studer, York Sure, Raphael Volz
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
{studer,sure,volz}@aifb.uni-karlsruhe.de
OntoWeb Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2000-29243.

Vrije Universiteit Amsterdam (VU)-Coordinator
Faculty of Sciences,
Division of Mathematics and Computer Science
De Boelelaan 1081a
1081 HV Amsterdam, the Netherlands
Fax and Answering machine: +31-(0)84-872 27 22
Mobil phone: +31-(0)6-51850619
Contactperson: Dieter Fensel
E-mail: dieter@cs.vu.nl
## Revision Information

<table>
<thead>
<tr>
<th>Revision date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-12-08</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>
Abstract

The OntoWeb Community ranges from academic to industrial partners who share an interest in the Semantic Web. Goal of WP 6 is to build a portal for this community serving as a platform for communication between partners and also between partners and other members of the World Wide Web. The portal is structured according to an ontology which serves as a share basis for supporting communication between humans, machines. This deliverable 6.2, presents a browsable prototype of the OntoWeb portal. Like other Semantic Web Portals, it is based on ontologies. The general goal of our approach is a semi-automatic construction of a community portal using the community’s metadata to enable information provision, information querying and information browsing of the portal. In this browsable prototype, AIFB provides a publish/review mechanism for information provision to the portal. STARLab provides an ontology-base querying and browsing system to the portal.
## Content Table

1. Creation of a browsable prototype of the portal ................................................................. 1
2. OntoWeb Consortium ..................................................................................................... 2
3. Revision Information ..................................................................................................... 3
4. Abstract .......................................................................................................................... 4
5. Content Table ................................................................................................................ 5
6. 1. Introduction .................................................................................................................. 6
7. 2. Architecture ............................................................................................................... 6
8. 3. Information Provision .................................................................................................. 7
9. 3.1 Information Provision via templates ........................................................................ 7
10. 3.2 Information provision via content syndication ....................................................... 8
11. 3.3 Publishing workflow ............................................................................................... 8
12. 4. Information Access .................................................................................................... 9
13. 4.1 The data store for ontologies and instances ............................................................. 9
14. 4.1.1 Schema for ontologies .................................................................................... 10
15. 4.1.2 Schema for instances ....................................................................................... 10
16. 4.2 Ontology-based querying and browsing Mechanism ............................................ 11
17. 4.3 Comparison with traditional querying engine ....................................................... 14
18. 4.5 Features Summary ................................................................................................. 14
19. 5. Portal Roadmap ......................................................................................................... 15
1. Introduction

The aim of WP6.2 is to create a 1st version of a browsable prototype based on an ontology seed from WP6.1 for OntoWeb portal. During the development of this prototype, AIFB and STARLab had set up an integrated approach, i.e., AIFB provides mechanism for information provision and STARLab provides mechanism for information access. The theory behind these implemented mechanisms is that both information provision and information access need to be ontology-based. For information provision, AIFB provides a publish and review mechanism that allow members of community to create their private contents and publish it after it is reviewed by a reviewer. On the other side, STARLab exploits semantic structure of ontology, provides an ontology map and navigational structures automatically in OntoWeb portal, allow community users to query and browse information based on community ontology. To allow community users access information transparently and easily, complex queries are hidden behind hypertext links, instances data is linked and integrated together by their semantic relationships in the portal.

2. Architecture

The current architecture of OntoWeb portal is shown in figure 1.

![Figure 1. Current architecture of OntoWeb](image-url)
The whole OntoWeb portal contains several components. These includes participants’ websites, tools for information provision or access and database to keep ontologies and instance data.

3. Information Provision

General goal of information provision is semi-automatic construction of a community portal using the community’s metadata. Information can be obtained from inside the portal that is using templates generated from an community ontology. Or it can be obtained from outside of the portal, i.e., collect instances based on community metadata located at external pages of community participants.

3.1 Information Provision via templates

Currently not enough instances based on metadata available, thus provision of instances based on metadata via templates is necessary. Users can edit and view (new) facts using templates. As in figure 2, these templates are generated based on metadata (ontology) in OntoWeb ontology base. Then community users can fill in data in templates and submit to a ZOPE server, this ZOPE server then store these instance data into ZODB and into the instance base in STARLab DOGMA server. Fro the 1st release, the ontology used to generate templates is not generated from an ontology base yet but from a ZODB, and also instance data is temporarily only saved in ZODB. In milestone 3 of this work package, the ontology will need to be fetched from the ontology base in the STARLab DOGMA server and instance data will be need to saved in the instance base in STARLab DOGMA server as well. This depends on a conversion to be implemented (responsible: AIFB).
3.2 Information provision via content syndication

The portal must allow centralized access to distributed information on participants’ sites. Participants can enrich resources located outside of the portal with metadata. This is done by syndicating information from participants by replicating their instance data. As in figure 3, both ontology and instances can be syndicated into the ontology base and the instance base in STARLab DOGMA server.

For the 1st release, similar to information provision via templates, that is, both ontologies and instances are only saved in ZODB instead of the ontology base and the instance base in STARLab DOGMA server, in milestone 3 of work package 6, ontology will need to be saved into the ontology base in STARLab DOGMA server, and instance data will be need to be saved in the instance base in STARLab DOGMA server also. Still, this depends on a conversion to be implemented. (responsible: AIFB)

3.3 Publishing workflow

Members can have a reviewer role. Normally members have private content on their web space; to make this information public, this content has to be reviewed by reviewers. The
whole process goes as in figure 4. When a user creates a content object, its status is private at the beginning. If the user wants to publish his contents, then he clicks on ‘Submit’, the status of the content becomes pending. On the other hand, on the reviewer side, the reviewer sees a pending publication waiting for his review, then he can choose either ‘publish’ or ‘reject’ depends on his judgment. In case of choose ‘publish’, the content was published and the content status becomes published.

![Figure 4. Publish and review workflow](image)

4. Information Access

4.1 The data store for ontologies and instances

In order to keep both metadata and instance data, an ontology base and an instance base have been set up in STARLab DOGMA server. A unified schema and two tables has been designed for representing both ontologies and instances. In such schema, as in table 1 or table 2, the first column is always a context label to identify which ontology we mean, the remaining three columns are similar to a RDF statement that contains three triples, i.e., a subject, a predicate and an object.
4.1.1 Schema for ontologies

Table 1 illustrates how ontologies are represented based on such schema in STARLab DOGMA server.

<table>
<thead>
<tr>
<th>Context</th>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoWeb</td>
<td>Student</td>
<td>is a</td>
<td>Person</td>
</tr>
<tr>
<td>OntoWeb</td>
<td>Student</td>
<td>studiesAt</td>
<td>University</td>
</tr>
<tr>
<td>OntoWeb</td>
<td>Person</td>
<td>email</td>
<td>String</td>
</tr>
<tr>
<td>OntoWeb</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 1 Ontology table

The information represented in table above is identical to ontologies expressed in DAML+OIL language as below.

```xml
<rdfs:Class rdf:ID="Student">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#studiesAt"/>
      <daml:toClass rdf:resource="#University"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</rdfs:Class>

<rdfs:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#email"/>
      <daml:toClass rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</rdfs:Class>
```

4.1.2 Schema for instances

Table 2 shows how instance data are stored in instance base.
OntoWeb: Ontology-based Information Exchange for Knowledge Management and Electronic Commerce

### Context

<table>
<thead>
<tr>
<th>Context</th>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>OntoWeb</td>
<td>Zheng</td>
<td>is_instance_of</td>
<td>Student</td>
</tr>
<tr>
<td>OntoWeb</td>
<td>Zheng</td>
<td>StudiesAt</td>
<td>VUB</td>
</tr>
<tr>
<td>OntoWeb</td>
<td>Zheng</td>
<td>Email</td>
<td><a href="mailto:zheng.jijuan@vub.ac.be">zheng.jijuan@vub.ac.be</a></td>
</tr>
<tr>
<td>OntoWeb</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Table 2 Instance table**

The information represented in table above is identical to instances expressed in DAML+OIL as below.

```
<Student rdf:ID="Zheng">
  <studiesAt>
    VUB
  </studiesAt>
  <email>
    zheng.jijuan@vub.ac.be
  </email>
</Student>
```

### 4.2 Ontology-based querying and browsing Mechanism

As in figure 5, the query mechanism is whenever a user submits a query in OntoWeb portal, he always submits two kinds of data together to query engine, one is an instance data, the other is the metadata of the instance data which tells the semantic of the instance data. The query engine will then be able to look up both ontology base and instance base with these two kinds of data to locate exact instances the user means.

The detail querying goes as follows: when the user goes to the home page of OntoWeb portal – http://starp14.vub.ac.be/, he can select a concept (subject) from an ISA hierarchy of the OntoWeb ontology and start an ontology-based query. As an example shown in figure 6, suppose he wants to look up a person in community, he selects ‘Person’ in ontology hierarchy, and then clicks on it. Then on the right side, a set of relations related to the concept of ‘Person’ appears as in Figure 2, now he can start to configure the query, choose and/or radio buttons and fill in the query strings in corresponding text field, finally click ‘Check’ button to submit his query to the query engine.
In this case, the query engine returns only one instance, a full professor ‘Robert Meersman’ as in figure 7. In some other cases, more than 1 instance could be returned; this also depends on how many instances meet supplied searching criteria.

Several properties of the instance “full professor Robert Meersman” are listed on the right side of the window as querying results. Some objects are displayed in hypertext links, this
indicates those objects are also resources having their own properties as well. Others are not, just static string, this means they have no any property associated with them as defined in ontology. Clicking on either one of objects with hypertext links, we will then go to another instance that is semantically related to former instance by the relation defined in ontology. So, different hyperlinks point to other different instances. By this means, one can also start an ontology-based browsing session on the OntoWeb portal as in figure 8.

All of these capabilities are due to the fact that, in OntoWeb portal, not only instance data, but also metadata is supplied to describe the instances data and relations between them. Therefore, instances data are always expressed using community metadata, they are bridged, linked and integrated together in OntoWeb by their semantic relations as defined in ontologies.
4.3 Comparison with traditional querying engine

Compared to a traditional querying engine, an ontology-based querying engine needs high quality instances data based on metadata defined in an ontology, usually via manual annotation. Therefore, it is more expensive than a traditional querying engine using only data (without metadata), but on the other hand the “traditional” querying engines are limited to more explicit ways of navigating the data.

Figure 8. Ontology-based browsing mechanism

4.5 Features Summary

Features of this prototype can be summarized as follows:

- Knowledge is managed by using ontologies;
- Semantically related knowledge is woven together by relationships defined in ontologies;
- Users can extend their own ontology, the ontology engineer can then merge, align these
ontologies;

✓ It is a more intuitive way of querying and browsing on OntoWeb, since it is knowledge-based system.

5. Portal Roadmap

The general development roadmap for OntoWeb portal can be illustrated as figure 9, it contains 5 milestones and 5 deliverables, i.e.

✓ Creation of a seed ontology for the portal, finished in time as September, 2001.
✓ Creation of a browsable prototype of the portal, finished in time as December, 2001.

➢ Complete ontology and portal, expect to be finished in June, 2002.
➢ Extended version of the portal, expect to be finished in March, 2003.
➢ Content upkeep, expect to be finished in May, 2004.

Figure 9. Development roadmap for OntoWeb portal