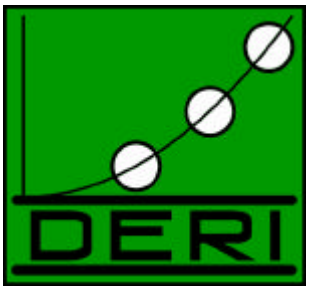

Requirements Analysis

Semantic Web Portal Project

swportal

Jos de Bruijn, Ying Ding, Ina O'Murchu

January 2004

Deliverable:
D3

Version:
0.2

Date:
8-Mar-04

DERI Ireland
University Road
Galway
IRELAND
www.deri.ie

DERI Austria
Technikerstrasse 13
A-6020 Innsbruck
AUSTRIA
www.deri.at

Table of Contents

1 Executive Summary

2 Introduction

This report provides a requirements analysis, based on the mission statement for the Semantic Web Portal project, which is:

> *It is our mission to create a Semantic Web Portal, demonstrating the maturity of Semantic Web technology in a real application. The technology, which we will develop together with industrial partners, during the course of the project, will be applicable to community portals for different communities.*
>
> *The two pillars for the portal will be the usability of the portal, both for inexperienced and expert users, and the support of communities, enabling cooperation within and between networks of people. The community support in the portal will not only be support in the passive consumption of information, but also support in the active publishing and collaboration by community members, through consensual vocabularies. Within the portal, we aim to bring together networks of people and to facilitate collaboration between them.*
>
> *The use case for showcasing our Semantic Web Portal technology will be a community portal for the Semantic Web community, at semanticweb.org. We aim to bring together research groups, research projects, software developers and user communities in the Semantic Web area.*
>
> *Our technological mission is to create a satisfying ontology management environment needed for Semantic Web enabled community portals and to make extensive usage of Semantic Web technologies for enhanced information processing facilities and to create means for the Semantic inter-operation between different communities and even different Semantic Web portals.*

As we can see in the mission statement above, the two pillars for the SemanticWeb.org portal are (1) usability and (2) the support for cooperation within and between communities. A community here is not the semantic web community as a whole, but rather smaller communities concentrated around research groups, Semantic Web projects and Semantic Web products. One important aspect here is the aspect of user communities. The semanticweb.org portal can bring together the researchers, developers and users of Semantic Web-enabled applications.

This document aims to provide a requirements analysis, detailing both user requirements and technical requirements.

In chapter 3 we analyze the user requirements for the portal with respect to the usability and collaboration support and provide a set of usage scenarios and use cases. Chapter 4 analyzes the technical requirements to make the semanticweb.org portal a true Semantic Web portal. These technical requirements follow from the user requirements.

3 User Requirements

As seen in the previous chapter, the two pillars for the semanticweb.org portal are the usability of the portal and the support for collaboration within and between communities. In this chapter, we address these two pillars. First, we address the usability of the portal, by stating the main requirements on usability. Then, we address the collaboration support provided to the different communities by the portal. These two sections can be seen as general for any portal. We then describe a number of usage scenarios for the portal and a set of use cases.

3.1 Usability

Usability for web portals, and for web applications, has different dimensions. A web application does not only need to have a good User Interface design, as any application, but has some additional requirements because of the unique properties of the web client-server architecture.

Some important requirements for a usable portal are:

- An intuitive User Interface, which exposes all functionalities to the user in a clear way; the User Interface should be constructed from a functional perspective and not a graphical or a technical perspective
- In order to assure usability across different computer platforms, the main user interface for the portal should be web-based and should adhere to current standards in the presentation to the user. More specifically, the following standards should be followed to ensure access to the portal across platforms and by all groups in society:
  - XHTML (http://www.w3.org/TR/xhtml1/)
  - CSS (http://www.w3.org/Style/CSS/)
  - WCAG (Web Content Accessibility Guidelines; http://www.w3.org/TR/WAI-WEBCONTENT/)
- The portal should be available to people with low bandwidth connection, e.g. modem users, mobile phone users. This means that the images used on the portal should be restricted in size and XHTML pages should be constructed in a space-efficient manner.

These three points are very important to keep in mind when designing the User Interface for the portal.

3.2 Collaboration Support

The semanticweb.org portal aims to enable collaboration within and between communities, concentrated around the Semantic Web research area, industry and users.

Ontologies are used for the explication of terminologies used by communities and to enable exchange of information between individuals or communities using different terminologies, through ontology mediation.

It is an important requirement that different individuals and different communities can understand each other, even though they are speaking different languages and/or using different vocabularies.

The important concept here is the *community*. A community can be any group of individuals with a common interest, who wish to exchange ideas or experiences and who wish to cooperate. Communities can be concentrated around research topics, around certain products or projects, etc…

3.2.1 Exchange Inside Communities

Information exchange inside communities should be facilitated by the portal. Features that could contribute to such exchange are:

- Enabling creating new community and finding individuals with common interests.
    - The portal should provide support in setting up a community and setting up a vocabulary for that community. This vocabulary should be mapped to the portal ontology.
    - The portal should provide support in finding individuals with common interests, e.g. through ontology-based search of social networks or the navigation of social networks. Existing social networks (e.g. FoaF) should be integrated if possible.
- Enabling finding individuals to find interesting communities
    - The portal should enable individuals to search the social network to discover community memberships of individuals with common interests
    - The portal should provide ontology-based search of communities, taking into account different vocabularies
- Providing different communication features for communities:
    - Bulletin board (forum-style messages)
    - Chat room
    - Mailing lists (both private and public)
    - Personalized view

Using ontologies, the view of the information can be personalized.

3.2.2 Exchange Between Communities / Portals

Communication should be possible between different communities, possibly located at different portals and using different vocabularies. The portal should provide mediation services in order to allow translation between the different vocabularies.

Requirement for information exchange between community portals:

- Publishing the same information to different portals
- Viewing information from different communities and different portals

3.3 Scenarios

This section provides a description of several usage scenarios, which capture the functionality required from the portal, from a user point-of-view. Each scenario contains a sequence of actions a user performs at the portal, in order to achieve his/her goal.

3.3.1 Topic-driven search

Summary:

A user wants to search for people, events, products, etc. in his/her area of interest. This area of interest can either be defined implicitly, through the user profile, or explicitly, through parameterization of the search.

We distinguish to distinct cases:

1. The user is registered with the portal or any other registration service, with which the portal communicates. The user's interests are stored in the user's profile (i.e. persistent identity). The user can update the profile either via some special form or via refinements based on the search results.
2. The user is anonymous. In this case, the user's interests need to be completely filled in before the search.

The user can say up to what extent the search parameters should be extracted from the user profile. The user can always add specific interests to the search parameters. These specific interests can also be easily added to the user profile.

The search is done over all the content on the portal and the content of any linked portals, as well as the content of mailing lists.

After a search is performed, the user can refine the search based on the search results and based on additional parameters entered by the user.

Details:

User Identification

We distinguish three different cases for user identification:

i) The user logs in to the portal using a username and password obtained from the portal.

ii) The portal received the user authentication from a 3rd party portal, known to the portal. The user had logged in to that third party portal beforehand.

iii) The user remains anonymous and the identity of the user is only maintained during the browsing session.

Search

We distinguish three different dimensions for the searching problem:

i) *Topics of interest* to the user (e.g. Semantic Web, Logic Programming, Databases, etc).

ii) *Categories of content* on the portal and other sources (news; event; person; etc.)

iii) *Content sources* used in the search (e.g. portal content, forums, mailing lists; publications, other portals, etc). This dimension is important, because depending on the source, the cost (in both time and money) can differ per source.

The user profile is taken as a starting point for the search query. The user can view the query from these different dimensions and select/deselect topics, categories and sources.

The user can enter the search from several different angles, based on the dimensions mentioned above:

1. When the user is viewing content on the web site, the user can immediately perform a search based on the topics of the content being view. For example, when the user views a news article about certain topics, the user can click the link: *search for related content*.

Another way to do a topic-based search is to explicitly select the topics to search for. Either by saying *display items of interest* or by using the link *perform topic-based search*. In the latter case, the user selects topics from a hierarchical view or searches for topics using a keyword-based search. In this case, also synonyms of topics are taken into account.

2. When the user is viewing content in a certain category, the user can search based on that specific category. For example, when the home page of the portal has a section containing the latest news, the user can click on the link *view all news of the last week* or the link *view related news.*
3. When the user is viewing content from a certain content source, for example a mailing list or the publications area, the user can choose to search inside this source. For example, when browsing a mailing list, the user can click the link *search for related threads in all lists.*

Refine search

i) Once the results of the search are returned to the user, the user can refine the query based on properties of the search results and using again the three dimensioned outlined in the previous step.

ii) Properties of the search results can also be used to update the user profile. If certain common properties among these results are identified, the user can choose to add these to his/her profile.

The user can refine the search query in different ways, based on the search dimensions outlined above. The user can add or remove topics or restrict searches to sub-topics. The user can broaden the search by selecting additional categories of content to search in or narrow the search by removing categories from the search query. Similarly for the content source.

Storing the query and/or answers

The user can store the query and/or some of the query answers as part of its profile.

i) If the user indicates that a certain content item is of interest to him/her, the user profile gets updated automatically, based on the content of the item.

ii) If the user indicates that a certain query is of interest to the user, the profile is also updated automatically.

iii) The user can specifically store a content item or a query for later review in the user's "items of interest".

Log out

The user leaves the portal, either by doing an explicit logout or by leaving the web site.

3.3.2 Information push

Summary:

- User logs into Portal and connects to their profile online.
- User connects to their community of practice and does a search for papers on relevant topic
- User searches for any announcements for conferences or a call for papers in the community
- User receives emails announcing updates on latest announcements to users profile email Inbox

---

Details:

-User logs on to portal and logs in to their profile the profile being specific to that user.

-User connects on the community of Interest that is within their profile that they have joined or wish to join

-User uses the community to search for information or to announce new details.

-Personalized email comes from users profile to community announcing a new conference

User is able to locate information on conference announcements more easily as information is held in one centralized place in users profile under communities user has subscribed to

User can subscribe to mailing lists that make such said announcements by subscribing to them online amongst the relevant communities of interest

User can search emails that have been posted to the community to trace previous discussions on areas of research interest

Search

User uses the various postings to community board to search for all relevant information using text based searching or an Ontology based search to list all areas of interest or relevance.

All topics or communities that user subscribes to send emails to user to announce new conference calls for paper or areas of research interest user as requested by user. User can gather the information from emails sent from community in portal to either their Inbox in Portal or own personalized email account outside of portal depending on option selected

User can aggregate all relevant information to either Inbox .

User can then monitor area of research through the relevant flow of emails that user has subscribed to receiving relevant information in the form of emails or through the relevant searches online.

Area of Interest can also be monitored through the use of RSS feeds that can be set up to aggregate information to users personalized Inbox outside of portal.

RSS feeds push relevant information to user.

3.3.3 Goal-driven search

Summary:

- User log on to his session with personalized information.
- Possible functions to fulfill user needs by portal:
    - User post his question to chat room, bullet board
    - User using keyword searching engine to search the portal.
    - User using ontology-based searching engine to search the portal (ontology browsing)
    - Using WIFI or FOAF to find related peers
- User log out of the session
    - User can store this information in his own session

Details:

User Log on

1) It is not necessary for user to log on to his session or personalized interface. But if he prefers, he can do that. In his personalized session, he can have his preferred topics and some other information stored somewhere.
2) The user log in with his user name and password.
3) The user remains anonymous and the identity of the user is only maintained during the browsing session.

User chooses to post his question to chat room, bullet board and mailing list

- *Chat room*

1) User goes to the chat room to post his questions.
2) User gets reply from the people who are currently in the chat room. If the answer satisfies the user, he stores this answer in his personalized session and log out. If not, he continues to try other functions provided by the portal.

- *Bullet board*

1) User goes to the bullet board to post his questions.
2) User gets reply from the bullet board. If the answer satisfies the user, he stores this answer in his personalized session and log out. If not, he continues to try other functions provided by the portal.

- *Mailing lists*

1) User goes to the online mailing lists of the portal to post his questions.
2) User gets reply from the mailing lists. If the answer satisfies the user, he stores this answer in his personalized session and log out. If not, he continues to try other functions provided by the portal.

User chooses the searching function of the portal

- Keyword-based searching

1) User keys in the keywords of his question into the keyword-based searching engine.
2) This search engine will search the information hosted by the portal (all the mailing list archives, all the posted information to the portal, all the linked webpages by the portal, etc.)
3) User gets the answer. If the answer satisfies the user, he stores this answer in his personalized session and log out. If not, he continues to try other functions provided by the portal.

- Ontology-based searching

1) user browse the ontology library of the portal to identify the related concepts or relations to his question
2) If he finds the related concepts, he can browse into deeper level and to the end finds the information he wants. If the answer satisfies the user, he stores this answer in his personalized session and log out. If not, he continues to try other functions provided by the portal.
3) If he cannot find the related concepts, he tries the ontology-based searching interface, which is the form-based where the instance of the concept, relation and attributes can be filled in. Then this search engine will search the information hosted by the portal (all the mailing list archives, all the posted information to the portal, all the linked webpages by the portal, etc.)
4) User gets the answer. If the answer satisfies the user, he stores this answer in his personalized session and log out. If not, he continues to try other functions provided by the portal.

Log out

1) User gets the right answer and saves it in his own personalized session.
2) If user log on from the beginning by using his username and password, then he should log out

3.3.4 Publishing Content

Summary:

A user wants to publish content on the portal. This content can consist of news items, calendar items, publications or any other type of content.

We distinguish here between content of general interest and very specific content. Content of general interest is of potential interest to a large segment of the portal users. The very specific content is only of interest to a small segment of the portal users. This smaller segment is often, but not always, organized in a community.

There are several kinds of content publishing, which we do not take into account, namely the publishing that happens outside of the portal. Example are sending emails to a mailing list, publishing content on different portals and scientific publications.

Details:

User Identification

We distinguish three different cases for user identification:

i) The user logs in to the portal using a username and password obtained from the portal.

ii) The portal received the user authentication from a 3rd party portal, known to the portal. The user had logged in to that third party portal beforehand.

iii) The user remains anonymous and the identity of the user is only maintained during the browsing session.

Annotating Content

Analogous to the search dimensions in the scenario 'Topic-driven search', the user has to take into account three dimensions for publishing content on the portal. In fact, the user needs to relate the content to certain topics, known in the vocabulary of the portal. Furthermore, the user needs to explicate the category to which the content belongs (e.g. news, events). Finally, the user needs to indicate to which content source the content is to be published (e.g. main portal content, a specific community on the portal, etc…).

Again the three types of information to be provided for the content item:

i) *Topics* (e.g. Semantic Web, Logic Programming, Databases, etc).

ii) *Category of the content* (news; event; person; etc.)

iii) *Content source* to be used for publishing the content item (e.g. portal content, forums, community, etc).

The user profile is taken as a starting point for these types of information. Especially when the user often publishes content about certain topics, in certain categories or to certain content sources, this information will be used as a starting point.

Furthermore, the recent actions of the user on the portal can be used for easy publishing. For example, when a user is browsing content in the news category, the user can publish news items with one click on a button. The same goes for browsing or searching certain topics and content sources.

To publish a document the user has to annotate the content with the three types of information indicated above. As was demonstrated, a lot of this annotation can be done automatically.

Not every user can publish to every content source. This depends very much on the user role. For example, not every user can publish news items on the front page of the portal and some communities might not allow publishing information in that community by non-members.

Publishing Content

After having chosen to publish content, the user is presented with a form to fill in the content to be published. The make-up of the form depends on the topics, content category and content source. For example, a news item needs different information than information about a person.

In many cases, the user is provided with the possibility to upload content, for example, pictures for persons or news items and PDF documents for publications.

Moderation

Besides the different user roles, some content sources might require moderation by a dedicated moderator or by community members. When content is moderated, we distinguish two distinct cases:

i) The content is *published*, after which it is moderated

ii) The content is *not published* immediately, but awaits moderator approval

For moderation we distinguish:

- The content is not moderated
- The content is moderated by a dedicated moderator, who decides to allow or disallow publishing the content
- The content is moderated by a group of moderators, who rate the content item. Based on the rating, the content item can be published (with or without the rating) or removed. The rating can also be used to emphasize the content items with higher rating and make the items with a lower rating less visible[1].
- The content is moderated by the user community. Users of the portal can rate the content item. Examples of this approach are Amazon (http://www.amazon.com/), where users can rate products for sale, and IMDB (http://www.imdb.com), where users can rate films. This rating is shown along with the content and helps users to distinguish higher-quality from lower-quality content.

Log out

The user leaves the portal, either by doing an explicit logout or by leaving the web site.

3.3.5 Updating Community Descriptions

Summary:

A user wants to update the vocabulary used by a certain community. It is possible that the vocabulary used by the community is missing some terms or that the community has evolved; this change needs to be reflected in the vocabulary used by the community.

[1] This technique is used by SlashDot (http://slashdot.org/). User comments are moderated by a large distributed group of moderators; comments with a high rating are shown directly on the page; comments with a lower rating can be reached through a hyperlink.

The vocabulary is not maintained centrally, by one person or organization, but distributed, by all community members, enabling as evolving vocabulary, which reflects an ever evolving community.

Details:

User Identification

We distinguish three different cases for user identification:

i) The user logs in to the portal using a username and password obtained from the portal.

ii) The portal received the user authentication from a 3rd party portal, known to the portal. The user had logged in to that third party portal beforehand.

Locating the Community

The user first of all has to locate the community of which to evolve the vocabulary. This can be done trough searching or browsing (cf. 3.3.1 Topic-driven search). In general, the user would be a member of the community and thus and browse to the community simply from the home page of the portal, which presents a personalized view to the user, based on the user profile.

Editing the Vocabulary

In the community page, the user can directly click the button 'Edit vocabulary' and start editing the vocabulary (cf. scenario 6.1), either via a web-based interface or a thick client with a richer user interface.

Moderation

After the user commits the changes, the updates to the vocabulary enter a process of moderation. As with moderation of content in the scenario 3.3.4 Publishing Content, we distinguish two distinct cases for moderation of updates in a community vocabulary:

i) The updates are *committed*, after which they are moderated

ii) The updates are *not committed* immediately, but await moderator approval

For moderation we distinguish:

- The updates are not moderated. This is not a very likely option for community vocabularies, since the vocabulary is used by a potentially large community.
- The updates to the vocabulary are moderated by a dedicated moderator, who decides to allow or disallow committed the updates.
- The updates are moderated by a group of moderators, who collaboratively decide to either commit or discard the updates. We current feel that this is the most likely option for the moderation of a community vocabulary.
- The content is moderated by the community. Community members can advise to either commit or discard the updates. These advise are taken into account be an administrator, who takes the final decision.

Log out

The user leaves the portal, either by doing an explicit logout or by leaving the web site.

3.3.6 Community Building

Summary:

Scenario 6.1: Collaborative Ontology Development

- User log on to his session with personalized information.
- User goes to ontology editor and develop it alone
- Many users from different locations go to ontology editor and develop it together
- The developed ontology has to be quality controlled by the domain expert. Once it passes the quality control, it will be published in the ontology libraries
- User saves the result and logout

Scenario 6.2: Relations between communities and portals

- User log on to his session with personalized information.
- User has list of interesting groups within the community portal to join

- User can open his own interesting group if he can have more than 10 people joining this group.
- Everyday portal is checking the statistical log of the activities of the interesting groups, if some of them have no activity for more than 30 days, these groups will go automatically to the archive.
- User in his interesting group can chat with the person in the group and posting email to mailinglist and browse and search mailinglist archive.
- User saves the result in his profile and logout

Details:

Scenario 6.1

User Log on

1) It is not necessary for user to log on to his session or personalized interface. But if he prefers, he can do that. In his personalized session, he can have his preferred topics and some other information stored somewhere.
2) The user log in with his user name and password.
3) The user remains anonymous and the identity of the user is only maintained during the browsing session.

User goes to ontology editor and develop it alone

1) User goes to the ontology editor.
2) User starts to search ontology libraries to find similar ontologies, user finds one which is exact what he needs. Then he saves the result and logout. If not,
3) User finds one ontology where he can start to add something more to generate his own ontology. Once he finishes, he saves the result and logout. If not,
4) User generates ontology from the scratch. Once he finishes, he saves the result and logout.

Many users from different locations go to ontology editor and develop it together (synchronous)

1) Users go to the ontology editor.
2) Users start to search ontology libraries to find similar ontologies, users find one which is exact what they needs. Then they save the result and logout. If not,
3) Users find one ontology where they can start to add something more to generate his own ontology. Once they finishes, he saves the result and logout. If not,
4) Users generate ontology from the scratch. Once they finish, they save the result and logout.

The developed ontology has to be quality controlled by the domain expert. Once it passes the quality control, it will be published in the ontology libraries

1) The developed ontology has been passed to domain expert for review
2) Once the domain expert approves this ontology, ontology will be published in the ontology library.
3) If the domain expert does not approve this ontology, the editors of this ontology will be informed together with the comments from the domain expert. Users can refine the ontology based on the comments and resubmit it to domain expert again, until domain expert agrees. Otherwise, users can just give it up.

User saves the result in his profile and logout

Scenario 6.2

User Log on

1) It is not necessary for user to log on to his session or personalized interface. But if he prefers, he can do that. In his personalized session, he can have his preferred topics and some other information stored somewhere.
2) The user log in with his user name and password.
3) The user remains anonymous and the identity of the user is only maintained during the browsing session.

User has list of interesting groups within the community portal to join

1) User goes to the session of interesting group
2) User browses the list of the interesting group to join
3) User joins one interesting group (to chat in the chat room, to post email to the mailinglist and search the mailinglist archives)

User can open his own interesting group

1) User goes to the session of interesting group
2) User opens one interesting group
3) User invites people to join this interesting group

Everyday portal is checking the statistical log of the activities of the interesting groups, if some of them have no activity for more than 30 days, these groups will go automatically to the archive.

1) Portal is check the activities of each interesting group at certain time of the day
2) Portal identifies some interesting groups which are not functioning for more than 30 days
3) Portal moves these portals to the archive.

User saves the result in his profile and logout

3.3.7 Content Maintenance

Summary:

- administrator loges on
- administrator update the content of the portal online
- administrator update the structure of the portal online
- old events will be updated and moved to the archive

- administrator loges out

Details:

Administrator log on

1) Administrator log on and he has highest level of access right.

Administrator updates the content of the portal online

1) Administrator has the highest right to update all the content of the portal.
2) All the updates can be made within the administrator account and online
3) Log file will be generated automatically to document all these changes

Administrator updates the structure of the portal online

1) Administrator has the highest right to update the structure of the portal.
2) All the updates can be made within the administrator account and online
3) Log file will be generated automatically to document all these changes

Old events will be updated and moved to the archive

1) Portal will check the date of the published events
2) Portal will move the out-of-date events to the archive

Administrator logs out

3.4 Use Cases

Based on the scenario description of the previous section, we have distinguished a list of Use Cases for the Semantic Web Portal. The section this list of use cases, along with a short description for each use case.

3.4.1 Login

A user logs in to the portal in one of the following ways:

- Registering as a new user
- Entering the username and password for registered users
- Redirect from a 3rd party identity provider
- Anonymous login. In this case the user has no persistent identity; the identity is only maintained during the session.

3.4.2 Browse

The user is presented with a personalized view of the portal, based on the user profile. The user can browse through different content categories (e.g. news, events), different topics (e.g. Semantic Web, Databases) and different content sources (e.g. community forums, portal news).

3.4.3 Search

The user can search through the portal and other content sources. The search query is specified in three dimensions:

i) *Topics of interest* to the user (e.g. Semantic Web, Logic Programming, Databases, etc).

ii) *Categories of content* on the portal and other sources (news; event; person; etc.)

iii) *Content sources* used in the search (e.g. portal content, forums, mailing lists; publications, other portals, etc). This dimension is important, because depending on the source, the cost (in both time and money) can differ per source.

3.4.4 Publish

The user can publish content in different sources, based on the user permissions. For example, communities might block publishing of content from non-members and content on the front page of the portal may only be published by a limited number of users.

Besides different *sources*, the user can publish in different *categories* and annotate the published content with different *topics*.

3.4.5 Edit Ontology

Community vocabularies can be created and updated using a web-based or a thick-client ontology editor. Changes to an ontology are usually submitted for moderation by a (group of) moderator(s).

3.4.6 Moderate Ontology Edits

Updates to ontology can be moderated by a single administrator, by a distributed group of moderators or by a user community.

3.4.7 Moderate Content

Content (to be) published can be moderated by a single administrator, by a distributed group of moderators or by the user community.

3.4.8 Update User Profile

The user profile can be updated in order to reflect publishing or editing actions of a user or to reflect topics of interest to a user.

3.4.9 Logout

After a browsing session on the portal, the user can either explicitly or implicitly log out (by exiting the portal).

4 Technical requirements

This chapter details requirements on the underlying system technologies and Semantic Web technologies for a Semantic Web portal.

Section 4.1 details requirements on underlying system technologies, such as database systems and operating systems. Built on top of these underlying system technologies are the Semantic Web technologies. We detail requirements on Semantic Web technologies in section 4.2.

4.1 System technologies

4.1.1 Data Management

There are three different types of data:

- unstructured
- semi-structured
- structured

Typically, unstructured data is simply plain text, without any annotation.

Semi-structured data is data annotated with some known tags, while structured data is organized using a schema, typically in a database.

Storage for unstructured and semi-structured data is typically provided by a simple file system, such as EXT2 or NTFS. Although a file system does provide some features in the areas of security and data protection (using a journaling file system such as EXT3 or ReiserFS), file systems still lack the functionality provided by many modern database systems.

One important feature, still lacking in file systems nowadays is the capability to issue random queries and combining the results of different sources. These functionalities are provided for by database management systems, which usually use a dedicated language for the retrieving, updating and deleting information (using a so-called Data Manipulation Language) and the definition of the data schema to which the data must adhere (using a so-called Data Definition Language). For most modern (relational) database systems, SQL (Structured Query Language) fulfills the roles of both the data definition and the manipulation language.

Requirements for a data management solution are:

- Flexible control over user authentication and authorization
- The ability to handle both unstructured and semi-structured data
- A clear overview of the data present in the data store
- The ability to perform fast queries and fast updates

Requirements for the usage of a relational database or even an RDF store depend on the ontology management solution, to be chosen.

The ontology layer is located on top of the data management layer and therefore they are dependent on each other. The chosen data management solution must be supported by the ontology management solution.

Sesame, for example, uses an RDF Store, which layers on top of a relational database. In this case, the data management layer is restricted to using a relational database.

OntoEdit, together with Ontobroker, uses the file system for storing the ontologies and can use relational databases (or files residing in the file system) for storing instance data.

4.1.2 Server Software

In server software, there are three main components:

- The Operating System (OS)
- The Application Server (which runs the business logic and accesses the underlying data store)
- The Web Server (on which the portal is hosted)

All server software is run on a computer, which runs some Operation System. The two largest current server operating systems are:

- Linux (with many distributions)
- Microsoft Windows NT (with different versions)

The main difference between the two operating systems is the fact that Linux is free, open-source software and Windows is not.

The choice of the operating system is important for the other services to be run on it. The database server, application server and web server all need to be run on the operating system and must support the operating system.

With respect to the application server, it is necessary that the application server supports the chosen programming language and has the necessary extensions for using Web Services.

The web server must support the chosen programming language and must run on the chosen operating system.

4.2 Semantic Web technologies

Ontology is the back bone for Semantic Web; therefore how to manage ontology in efficient and scalable way is the central point to enable semantic web functions for the users. Here in our Semantic Web functional portal design. We should focus ourselves on the following key technologies mentioned immediately after.

4.2.1 Ontology Management

As the number of different ontologies is increasing, the task of maintaining and re-organizing them in order to facilitate the re-use of knowledge is challenging. A breakthrough in ontology technology would require methodological aids and tools that enable effective and efficient development. A key aspect in achieving this is successful re-use of ontologies. Being developed for supporting knowledge sharing and reuse, it is the lack of proper support of ontology re-use that hampers a broader dissemination of the ontology. *Ontology library systems* are an important tool in grouping and re-organizing ontologies for further re-use, integration, maintenance, mapping and versioning.

4.2.1.1 Ontology Creation

User friendly tool environments should be provided for mainly assisting the creation of ontologies. So main features could include:

- Graphical support for creating ontologies in the portal(such as WebOnto, Figure 1)

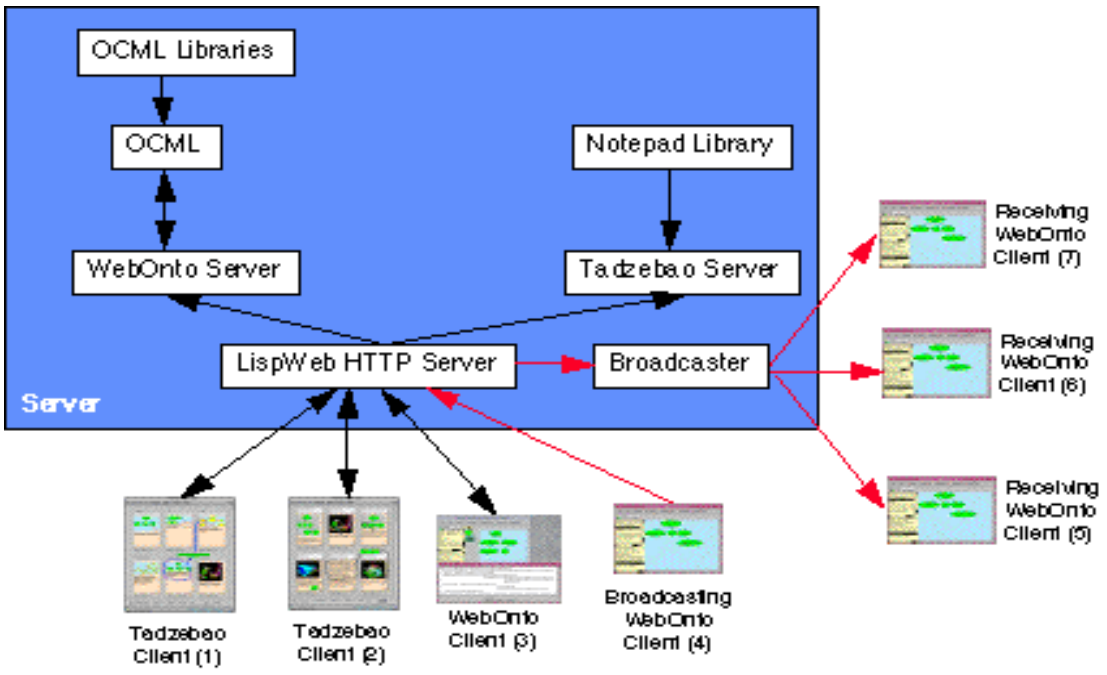

Figure 1: WebOnto: Graphical support for creating ontologies

- Support for both asynchronous and synchronous creation on ontologies. Portal users should be highly involved in creating ontologies, therefore to reflect the ontology as emergent semantics in the community. This is mainly for multiple users in different locations (WebOnto also supports this).
- Re-use of existing ontologies, eat your own dog foot and not to invent a new wheel again.
- Support extra ontology editor plug-in, make it more interoperable and flexible
- Providing some upper level ontologies so as to keep the modelling more consistent and smoothing, and later on easing ontology mapping and aligning
- Providing ontology templates, which is the same effort like one before to keep the consistency and ease the later on ontology mapping and aligning
- Providing help files and historical lists to the portal users, and also common questions and answers.
- Providing *common vocabulary* (or faceted taxonomy) for the portal. At any rate, it should eliminate the implicitness and misunderstanding of terms in different ontologies (due to synonyms, homonyms, etc.) for most generic classes. The structures of these common vocabularies or multiple controlled vocabularies must be faceted, or modulated so as to facilitate the re-use, mapping and integration of ontologies. These vocabularies can help in simple synonym matching, sibling analysis, and disjoint partition checking.

4.2.1.2 Ontology Maintenance

Ontology maintenance is normally the bottleneck, which contains many challenges nowadays in Semantic Web areas. Most of them have no solutions yet, therefore this makes the hardest part of our Semantic Web portal design.

Ontology maintenance contains:

- How to store the ontologies
    - Ontology should be web accessible, either client/server-based or decentralized architecture
    - It is necessary to *classify* ontology in order to facilitate better management of ontologies. Some of the ontology classification mechanisms available are based on distinguishable features of ontologies, such as: the

subject of ontologies (The DAML ontology library system); the *structure* of the ontology (The Ontolingua ontology library system); *stratified upper-level* ontology (ONIONS), *Modular organization* organizing units into modules (This serves to maximize cohesion within modules and minimize interaction between modules).

- Others:
  - Different *naming policies* assist the ontology library system to achieve the modular organization or stratified storage of ontologies.
  - The disjointed partitioning of classes can facilitate modularity, assembling, integrating and consisting checking of ontologies. Thus, the partition modification has proven to be extremely valuable for editing purposes.
  - Linking class names with their own contexts or using name space for differentiating them can serve to prevent violation within individual ontologies.
  - As ontologies continue to grow, so too does the importance of systematic and consistent naming and organizational rules.

- How to search ontologies

  - Browsing functions: a visualized browsing environment, using hyperlinks or cross-references to closely related information. Information visualization can be provided to the portal users.
  - Offering advanced searching features by adopting various existing information retrieval techniques, such as keyword based searching, wild-card matching, fuzzy matching, cited linking searching (like Google)
  - database searching features: if it is stored in database (most likely), multiple table searching, cross-field searching
  - AI heuristic techniques: some heuristic AI searching algorithms. Since ontologies are tree-structured, some tree based AI heuristic searching can be provide to support efficient searching (A* searching)

- How to edit or update ontologies

  - User friendly editing and maintaining environment: user should be able to add, delete and update any information in the ontology.
  - User should be divided into different user groups with different editing and updating rights, such as portal user can have different levels of access rights.
  - Quality control should be provided to let rubbish-in and rubbish-out

- Some level of reasoning support: consistency checking, diagnostic checking,
- Support for changes, and adaptation of ontologies for different applications
- Personalized editing and updating ontology environment: We can monitor the portal user profiles based on access patterns in order to personalize the view of ontologies.

- How to identify ontologies

  - Unique ontology URL, Identifier and name are used as the identifier for ontologies

- Reasoning support for ontology maintenance (mainly consistency checking)
  - A simple reasoning function should be included in order to facilitate ontology creation, ontology mapping and integration, such as consistency checking, diagnostic checking, and so on.
  - Reasoning should not take very long time since the portal user would like to have prompt reply.

- Other requirements:

  - **Explicit documentation**. Each ontology in an ontology library system should be extensively documented. The documentation should include such information as how the ontology was constructed, how to make extensions and what the ontology's naming policy, organizational principles and functions are. Such explicit documents about the ontologies themselves will pave the way for efficient ontology management and reuse.
  - **Ontology scalability**. Ontology library systems should also consider increasing the scale of ontologies.

4.2.1.3 Ontology Versioning

Versioning of ontology is extremely important due to the dynamic changes of the nature itself. Unfortunately, at the moment, even in the research areas, it is still the open question. Klein (2002) has identified the following properties of an version relations:

- Transformation or actual change: a specification of what h as actually changed in an ontological definition.
- Conceptual relation: the relation between constructs in the two versions of the ontologies: such as equivalence relations, subsumption relations, or logical rules
- Changes on descriptive meta data
- Scope: a description of the context in which the update is valid. Such as valid date or any other more extensive descriptions of the scope in various degrees of formality
- Conceptual change: a change in the way a domain is interpreted which results in different ontological concepts or different relations between those concepts
- Explication change: a change in the way the conceptualization is specified, without changing the conceptualization itself
- Packaging of changes: how to implement changes into an ontology. There are two dimensions: granularity and method (transformation, replacement and mapping)

If we try to simplify the question as just to be able to handle the changes in ontology by creating and managing different variants of it, some of the requirements are foreseen:

- Ontology identification and reference
- Classifying changes
- Stating the scope
- Identifying conceptual change or explication change
- Packaging of changes
- Good documentation
- Historical support
- Help files and questions and answers

4.2.1.4 Ontology Interoperation

Some requirements for ontology interoperability can be list as followings:

- Flexible for different ontology language, such as OWL, RDFs, Ontolingua, KIF, etc.
- Flexible for different plug-in of ontology editors
- Flexible for different plug-in of ontology reasoners

- Layered structures
- Dynamic component binding
- Meditation for mapping and aligning

Another important feature in ontology interoperation is to support ontology mappings. Requirements for that could be:

- Drag and drop interface support for manually mapping ontology
- Consistency checking for generated mappings
- Run-time instance transformation during the ontology mapping

5 Conclusion

6 Bibliography

[Ding et al., 2002] Ding, Y., Fensel, D. (2002): Ontology Library Systems. The key to successful Ontology Reuse. In: *Proceedings of the First Semantic Web Working Symposium.* California, USA, pp. 93—112.

[Han, 2003] Han, S (2003): *Commercial Portal Products*, Semantic Web Portal Project Deliverable 10, v0.1. *http://www.nextwebgeneration.org/projects/sw-portal/papers/deliverables/CommercialPortal.zip*

[Klein et al., 2002] Klein, Y., Fensel, D., Kiryakov, A., and Ognyanov, D. (2002): Ontology versioning and change detection on the Web. In: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*. Sig uenza, Spain.

[Lausen et al., unpub] Lausen, H., Stollberg, M., Lara, R., Ding, Y., Sung-Kook, H. and Fensel, D. (unpublished): *Semantic Web Portals – State of the Art Survey*.

[Maedche et al., 2003] A. Maedche and S. Staab and N. Stojanovic and R. Studer and Y. Sure (2003): *Semantic portal - The SEAL approach*, In *Spinning the Semantic Web*. D. Fensel, J. Hendler, H. Lieberman, W. Wahlster (eds.) MIT Press, MA, Cambridge.

[Reynolds & Shabajee] Reynolds, D. and Shabajee, P.: *Semantic portals demonstrator - requirements specification*, SWAD-Europe Deliverable 12.1.5, initial release. *http://www.w3.org/2001/sw/Europe/reports/requirements-demo-2/*