SEMANTIC LINKING AND PERSONALIZATION IN CONTEXT

Melike Şah

A thesis submitted for the degree of Doctor of Philosophy

School of Electronics and Computer Science,
University of Southampton,
United Kingdom.

June, 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

SEMANTIC LINKING AND PERSONALIZATION IN CONTEXT

by Melike Şah

The World Wide Web (WWW) is intended for humans to create and share documents. However, it does not support machine-processable data and automated processing. The Semantic Web is an extension to the WWW and can overcome its shortcomings. The Semantic Web provides the technology for creating and sharing data in machine-processable semantics. As a result, the data can be used and shared in effective ways between cross applications.

In this thesis, we investigate the Semantic Web technologies for context-based hyperlink creation and personalization. Two different contributions are presented using Semantic Web technologies. First, we introduce and implement a novel personalized Semantic Web-enabled portal (known as a semantic portal), which is called SEMPort with the aim of improving information discovery and information sharing using the Semantic Web technologies. We also provide different Adaptive Hypermedia (AH) methods using ontology-based user models. In our second contribution, we introduce and implement a novel personalized Semantic Web browser, called SemWeB which is a browser that augments Web documents with metadata. It creates and personalizes context-based hyperlinks and data using ontologies. We have also developed a new behaviour-based user model for Web-based personalization which supports different AH methods. In addition, a novel semantic relatedness measure is proposed.

The evaluations showed that our contributions to the development of hypertext systems using Semantic Web technologies are successfully applied for context-based link creation and personalization.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, *Melike Şah* declare that the thesis entitled

Semantic Linking and Personalization in Context

and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- part of this work have been published as:

Şah, M., and W. Hall (2007). *Building and Managing Personalized Semantic Portals*. In Proceedings of International World Wide Web Conference, 1227-1228.

Şah, M., W. Hall, N. M. Gibbins and D. C. De Roure (2007). *SEMPort – A Personalized Semantic Portal.* In Proceedings of the Eighteenth ACM Conference on Hypertext and Hypermedia, 31-32.

Şah, M., W. Hall and D. C. De Roure (2008). *Designing a Personalized Semantic Web Browser*. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS, 5149, 333-336, Springer-Verlag Berlin Heidelberg.

Şah, M., W. Hall and D. C. De Roure (2008b). *SemWeB: A Semantic Web Browser for Supporting the Browsing of Users using Semantic and Adaptive Links*. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems Doctoral Consortium, LNCS, 5149, 431-436, Springer-Verlag Berlin Heidelberg.

Şah, M., W. Hall and D. C. De Roure (2009). *SemWeB Semantic Web Browser – Improving Browsing Experience with Semantic and Personalized Information and Hyperlinks*. 1st National Symposium for Humanities and Technology.

Signed: *Melike Şah*

Date: 10/6/2009

Acknowledgements

First and foremost, I would like to thank my supervisor Prof Dame Wendy Hall for her time in revising papers and this thesis, and for her support, guidance and feedbacks throughout my PhD.

I would like to thank my supervisor Prof David De Roure for his suggestions and guidance; Dr Jing Zhou and Dr Tim Miles-Board for helping me to find my research path in the early stages of my PhD.

I would like to thank and acknowledge Dr Nicholas Gibbins for his feedback and answering my questions and helping me to convert the ACM Computer Classification System into RDF.

I would like to acknowledge Dr. Gary Wills, who answered my questions about evaluation methods and techniques. I also thank all of the participants who took part in the evalutions and Mr. Quintin Gee for proof-reading my thesis in its early stages.

I would like to thank my friends in IAM; Khana, Reena, Salma, Noura, Lailatul and Pei for their good company and who made my stay in the University of Southampton a rich experience. My special thanks go to Jane Morgan who helped me during my studies. In addition, I would like to thank Dr. Vania Dimitrova for her useful comments and suggestions about my work.

I would like to express my gratitude to all those who gave me the opportunity of completing my thesis and to the support of the EPSRC.

Finally, I would like to express my gratitude to my family and my fiancé, Cem, for always being there to support and believe in me, and their motivation throughout my PhD.

This thesis is dedicated to my family.

Definitions and Abbreviations Used

ACM	Association for Computing Machinery
ACM CCS	ACM Computing Classification System – is a subject classification system for computer science devised by the Association for Computing Machinery
AH	Adaptive Hypermedia
AHS	Adaptive Hypermedia System
ACS	Attribute Change Service – is a Java Servlet developed within SemWeB to edit attribute values from the user profile
ADS	Attribute Delete Service – is a Java Servlet developed within SemWeB to delete attributes from the user profile
AES	Add Expertise Service – is a Java Servlet developed within SemWeB to add expertise values to the user profile
AIS	Add Interest Service – is a Java Servlet developed within SemWeB to add interests to the user profile
AJAX	Asynchronous Javascript and XML
ANNIE	A Nearly New Information Extraction System – developed with GATE for information extraction
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CERN	International Laboratory for Particle Physics
CSS	Cascading Style Sheets
D2R Server	A tool for publishing relational databases on the Semantic Web
DBD	DBpedia Definitions – is a goal service provided by SemWeB for finding DBpedia definition of a semantic resource
DBLP	Bibliography database that provides metadata about scientific papers, conferences, authors, etc.
DBpedia	A database that provides metadata about Wikipedia resources
DL	Description Logic
DLS	Distributed Link Service – is a service implemented from Microcosm to incorporate links from wide range of network information
DOAP	A vocabulary for describing open-source projects

DOM	Document Object Model
DTD	Document Type Definition
ECS	School of Electronics and Computer Science
ECS Southampton	Provides metadata about people, publications, modules, etc. within the ECS
ECS CMWP	Course Modules Web Page of the ECS
FMLDB	Find More Links within DBpedia – is a goal service provided by SemWeB for finding related links from the DBpedia database
FOAF	Friend-Of-A-Friend – a vocabulary describing people, their activities and relationships to other people
FRESS	File Retrieval and Editing System – a hypertext system
FRP	Find Recent Publications – is a goal service provided by SemWeB for finding recent publications of a person from the DBLP domain.
FTP	File Transfer Protocol
GATE	A open-source text engineering architecture for extracting named entities from documents
Geonames	A database that provides metadata about geographic data
GUI	Graphical User Interface
HCI	Human Computer Interaction
HES	Hypertext Editing System – a hypertext system
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
Idf	Inverse Document Frequency – a text retrieval method
IE	Information Extraction
IES	Information Extraction Service – is a Java Servlet developed within SemWeB to perform information extraction and semantic annotation
IEEE PAPI	IEEE Public And Private Information for Learners – is a data interchange specification developed for communicating between different systems
IMS LIP	IMS Learner Information Package Specification – is a specification recording lifelong achievements of learners and transfer of these records between institutions
IR	Information Retrieval
JAPE	Java Annotations Pattern Engine for GATE
JAPEC	JAPE-to-Java Compiler for GATE

KB	Knowledge Base
KMS	Knowledge Management System – a hypertext system
LAWVC	Link Annotation With Visual Cues
LOM	Learning Object Metadata – is a specification for representing learning objects with metadata
N3	Notation 3 – a format for representing RDF triples
NLS	oNline System – the first working hypertext system developed by Douglas Engelbart and his team
NNTP	News Network Transfer Protocol
OWL	Web Ontology Language
POS	Part-of-Speech tagger
PR	Processing Resource – represents a component within ANNIE
PWW	People Work With – is a goal service provided by SemWeB for finding related people working on same projects within the ECS
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RIF	Rule Interchange Format
RS	Registration Service – is a Java Servlet developed within SemWeB to register users to the personalization
SKOS	Simple Knowledge Organization System – an area of work for developing specifications and standards to support the use of knowledge Organization systems (KOS) such as thesauri, classification schemes and taxonomies within the framework of the Semantic Web
SLS	Semantic Linking Service – is a Java Servlet developed within SemWeB to create semantic information and links from semantic instances
SRISW	Searching for Related Semantic Information on the Web – is a goal service provided by SemWeB for finding related links on the Web
Tf	Term Frequency – a text retrieval method
Turtle	A format for representing RDF triples
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	The World Wide Web Consortium
WWW	The World Wide Web
XML	eXtensible Markup Language

1 Introduction

The history of hypertext begins with an article titled “As We May Think” written by Vannevar Bush in the Atlantic Monthly, in 1945 (Bush, 1945). In this article, Vannevar Bush proposes ideas for a photo-electrical-mechanical device called *Memex*, which stores textual and graphical information. In this article, Vannevar Bush advocates non-linear access to information by means of associative indexing, since the human mind works with association. Bush also suggests the concepts of trails in Memex, where associated related information can be found using these trails. The trails are the first conception of hyperlinks.

The ideas of Vannevar Bush influenced the work of Ted Nelson and Douglas Engelbart. In 1965, Ted Nelson coined the word *hypertext* and *hypermedia* when describing his Xanadu system (Nelson, 1965). Nelson stated that hypertext means *“nonsequential writing – text that branches and allows choice to the reader”*. Links that connect different texts are called hyperlinks. Nelson also stated that *hypermedia* is as an extention of hypertext to include multimedia objects, such as sound, video and picture (Nelson, 1965).

With influence from the Memex, in 1968, Douglas Engelbart demonstrated the first working hypertext system, *oNLine System* (NLS). In this demonstration, Engelbart successfully showed cross-referencing and hyperlinking. Following this, many hypertext systems have been developed, some of which are described in Chapter 2.

Since its invention by Tim Berners-Lee in 1989 (Berners-Lee and Fischetti, 1999), the World Wide Web (also known as WWW or the Web) has become today’s most successful and widely used hypertext system and was a milestone in the development of the hypertext and hypermedia systems. The Web is essentially a network of documents interconnected by an unbounded number of hyperlinks. It is intended for humans to create and share information. It accomplishes this with human friendly data format (HTML[1]) and universal Internet protocols (http[2], nntp[3] and ftp[4]). However, the Web lacks from semantics, since machines cannot understand HTML and HTML cannot be shared between applications. To overcome the limitations of the Web, Tim Berners-Lee, James Hendler and Ora Lassila (Berners-Lee et al., 2001) introduced the Semantic Web, which is an extension of the Web to enable such information to be made understandable by machines using Semantic Web standards (e.g. Resource Description Framework (RDF)). Using Semantic Web standards, data can be accessed and processed automatically as well as shared across applications.

1.1 Motivation

Hyperlinks are first-order objects in a hypermedia system. They allow us to navigate the hyperspace and discover more information. However, there are limitations to links. Embedded hyperlinks within the Web page can be insufficient for navigating the hyperspace, since links can be expensive to create and maintain. This results in loosely created links between Web resources. The Semantic Web can be used to overcome the limitations of the Web using machine-processable semantics. Our aim in this research is to improve linking between Web resources by creating context-based hyperlinks using Semantic Web technologies.

The Web contains enourmous amounts of information and it is difficult for users to locate right information. Adaptive Hypermedia (AH) is an alternative to the traditional "*one-size-fits-all*" static hypermedia systems (Brusilovsky, 1996). User models – the goals, preferences, knowledge and interests of a user (or group of users) – are used to personalize the contents of the hypermedia systems. Although AH systems (AHSs) are very useful, the drawback is their closed architectures. These systems usually have their own formats for the modelling of the content domain, the user, and the delivery

---

[1] Hypertext Markup Language

[2] Hypertext Transfer Protocol

[3] News Network Transfer Protocol

[4] File Transfer Protocol

environments. This causes interoperability problems and the Semantic Web is a potential solution for these problems. The AH community has already taken advantage of the Semantic Web standards and generic user modelling standards have been developed, such as the IMS Learner Information Package Specification (IMS LIP, 2008) and the IEEE Public And Private Information for Learners (IEEE PAPI, 2008). The most commonly used properties in AH for personalisation is the users 'prior knowledge' of domain/subject area. This can be represented for example, as competencies in subjects (or skills), or knowledge of particular concepts. They are frequently used in adaptive elearning, adaptive information portals, adaptive tours, etc. In addition, both IEEE PAPI and IMS LIP standards are created for adaptive elearning applications and they use 'prior knowledge' of learners as the basis for the adaptation. However, they are not suitable for Web-based personalization since the interests, goals, and preferences of users are the mostly used user features in Web-based AHSs. In our research, we aim to develop Semantic Web-enabled personalization architectures, so that information can be adapted to individual users in different domains.

In the rest of this section, we discuss our motivations in the two areas that we have contributed to: semantic portals and Semantic Web browsers. Our most significant contribution is to the Semantic Web and AH. Our research did not specifically investigate the Human Computer Interaction (HCI) aspects.

1.1.1 Semantic Portals

In this research, we investigate the benefits of the Semantic Web technologies in the context of Web portals. A Web portal is a Web access point and it usually gathers information or collects Web pages into a single organized site. Conventional Web technologies are used in the implementation of a Web portal and these technologies have limitations; information access, search, integration and sharing are difficult and time-consuming tasks (Lara et al., 2004). To overcome the problems of conventional Web technologies, many Semantic Web-enabled portals, which are often called *semantic portals*, have been introduced. Examples of well known semantic portals are described in Chapter 5.

The problems associated with the existing semantic portals can be categorized under two groups: content edition/provision and information access. Semantic portals provide mechanisms for the edition and provision of the metadata of the contents. However, the process of content provision in semantic portals is a difficult task, especially where the content is continuously changing. The problems associated with content provision are twofold: the newly-added content through content provision interface cannot be seen without restarting the Web server, and Web interfaces are difficult to insert information into. On the other hand, in semantic portals, information can be accessed through search and navigation facilities. In most of these cases, navigation is not very effective; links to similar pages are often not presented and the same page content and the same set of hyperlinks are shown to all users. In this research, we aim to enhance existing semantic portals with context-based links, personalization and real-time easy-to-use content edition/provision Web interfaces.

1.1.2 Semantic Web Browsers

In our work on Semantic Web browsers, we extended the research we undertook on semantic portals. Semantic portals use databases to store the portal's ontologies and metadata. In addition, personalization is based on portal metadata and specific to the portal's domain. In this research we aim to bring Semantic Web metadata into everyday Web browsing using a Semantic Web browser so that context-based semantic hyperlinks can be added to the Web documents and users can be guided using personalized contents and hyperlinks.

With the development of the Semantic Web technologies and widely acceptance of Semantic Web standards, now open standard metadata is available on the Web, which is known as "*Linked Data*". Berners-Lee defines linked data as *"a method of exposing, sharing, and connecting pieces of data on the Semantic Web"* (Berners-Lee, 2006c). The linked data Web is the network of interconnected RDF nodes on the Semantic Web. The DBpedia (DBpedia, 2008) and the DBLP (DBLP, 2008) are two well known examples of linked data. In our opinion, linked data provides a rich source of information that can be utilized to create context-based hyperlinks on the Web documents. In this thesis, we use the terms *linked data* and *Semantic Web* with different meanings. For instance, linked data is used to describe the interconnected open standard metadata of the Semantic Web. The Semantic Web is used as a general methodological term that describes the technology for creating, publishing, linking and using the metadata.

Related work in this context particularly the development of linked data browsers is discussed in Chapter 5. Linked data browsers take the dereferenceable Uniform Resource Identifier (URI) of a resource and allow navigation and exploration of the resulting RDF graphs with user friendly Web interfaces. We take a different approach. Linked data browsers seperate metadata from the Web documents and they are browser for the linked data. In our opinion metadata could be more useful if it is presented within the context of the Web document, such as demonstrated by COHSE (Carr et al., 2001) and Magpie (Dzbor et al., 2003)). In our approach, we generate a Semantic Web browser for Web documents; we extract and annotate Web documents with metadata and provide semantic links within the page context. In this way, users are not required to adopt wholesale vision of the Semantic Web, but they are seamlessly supported by metadata in their everyday Web browsing. Futhermore, we aim to adapt information to the individual users in open-corpus Web content. With our approach, users are only required to log in to their Semantic Web browsers and they can be provided with AH on different Web sites. Thus, our contribution is to the open-corpus AHSs. Brusilovsky and Henze define an open-corpus AHS as an *"AHS which operates on an open-corpus of documents, e.g., a set of documents that is not known at design-time and, moreover, can constantly change and expand*" (Brusilovsky and Henze, 2007).

1.2 Significance of the Research

Our research is stimulated by the following motives:

- In a semantic portal, ontologies are used for sharing data between users. Ontologies can also be utilized to improve linking between resources. In this thesis, the benefits of using ontologies in a semantic portal are tested and demonstrated with a case study.
- Different users have diverse browsing needs and it is not reasonable to show the same set of hyperlinks and information to all users. This can be alleviated using AH. In our research, we experiment with ontology-based user models for tolerating data to individual users in a semantic portal and in a Semantic Web browser.
- Linked data is a new trend of open source metadata on the Web and its popularity is growing over time. We discuss and show how users can benefit from linked data without the wholesale adoption of the vision of the Semantic Web using a Semantic Web browser.

- Linked data provides new possibilities for achieving open-corpus AH by dynamically relating user models to any dereferenceable URI.
- Existing user modelling standards are mainly developed for learners and they are not suitable for Web-based personalization. New user models are required which can support browsing needs of users. We introduced a new behaviour-based user model for Web-based personalization in a Semantic Web browser.

1.3 Research Hypotheses

The hypotheses of the thesis can be stated as follows:

1. Semantic Web technologies can be used to generate rich ontology-based links between Web resources using ontologies.
2. Semantic Web technologies can overcome interoperability problems of AH using formal semantics (e.g. RDF).
3. Linked data is a new trend of open source metadata and it can be used to generate context-based hyperlinks within Web documents. This implies the following sub-hypothesis:
    a. Linked data can be used to annotate Web documents.
    b. Information from linked data can be searched combined and displayed using Semantic Web standards (e.g. RDF).
4. Linked data can be utilized to achieve open-corpus AH in a Semantic Web browser; annotated Web resources with deferenceable linked data URIs can be related to user profiles during browsing.
5. There is a need for a new user model which represents the user's browsing goals, interests and preferences to accomplish Web-based personalization.

1.4 Research Scope

Figure 1-1 gives a snapshot of the various technologies utilised in this thesis. In this thesis, we have two different contributions. One of them is to semantic portals and the other is to the development of Semantic Web browsers.

The thesis contributes to the semantic portals domain by providing semantic links and personalization. Personalization is achieved by using an ontology-based user model.

In the domain of Semantic Web browsers, the thesis adds a new behaviour-based user modelling architecture to a Semantic Web browser. In addition, a semantic annotation mechanism is employed to extract semantic metadata from ordinary Web pages. In this way, the thesis shows that open-corpus semantic linking and personalization can be achieved in different domains on the Web by using open standard linked data.

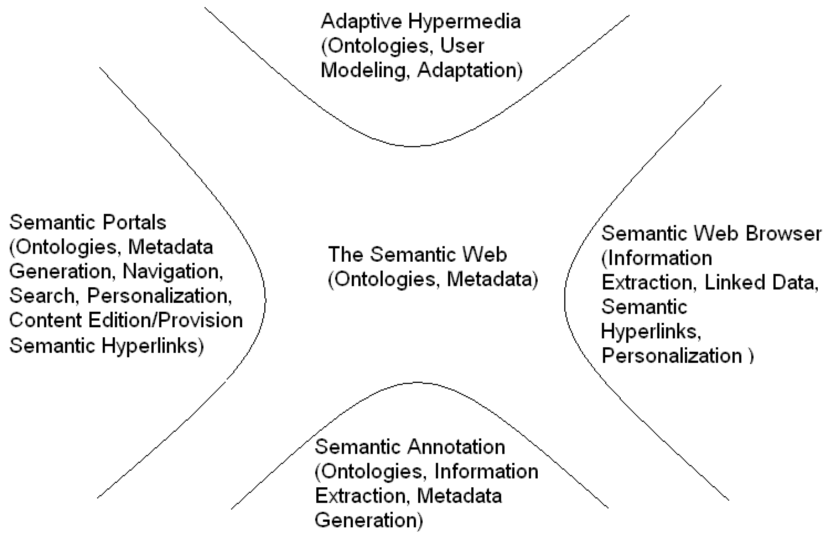


Figure 1-1 The research scope

1.5 Contributions

1.5.1 Building and Managing Personalized Semantic Portals

In our research, to address the problems associated with semantic portals, we propose an ontology-based semantic portal, which we call *SEMPort*. To alleviate the problems of the content editing/provision interfaces, we have developed an easy-to-use distributed Web interface for inserting, updating and removing instance attributes in real-time. In addition, the content of the portal is presented with personalized views depending on different users. Various personalized navigational support techniques are designed, for instance personalized homepages, personalized link sorting, and annotation of related links with visual cues. The personalization is performed through the use of an ontology-based user model, which collects information about the user.

To improve the browsing facilities for the users, the content is also enriched with automatically-generated context-based hyperlinks, which we call *semantic hyperlinks*.

These provide links to relevant pages and guide users to related items using four different kinds of link: explicit, inverse, implicit, and recommendation links. To facilitate information access, ontology-based search is also integrated into the semantic navigation. It is possible to perform concept-specific searches during the browsing.

SEMPort is implemented with reusable components to allow the development of other portals at a low cost. To illustrate our approach, we use the ECS Course Modules Web Page (ECS CMWP). This work has been published in (Şah and Hall, 2007) and (Şah et. al, 2007). In addition, a number of evaluations of our approach were performed; a structured review and an empirical study were undertaken.

1.5.2 Designing a Personalized Semantic Web Browser

In this research, we wanted to extend the research we have undertaken on semantic portals. Our aim is to use the Semantic Web for supporting browsing in different Web domains, using ontology-based hyperlinks and to personalize the information to the needs of the users. To achieve these aims, we developed a Semantic Web browser, called *SemWeB.* SemWeB uses linked data for generating context-based hyperlinks on Web documents and provides AH on different websites.

In our approach, we aim to provide semantic metadata in the context of Web pages. We used a standard Web browser, the Mozilla Firefox Web browser, and extended it with a sidebar. Users can use the SemWeB sidebar to access semantic hyperlinks, but first they need to annotate the Web page. To make this easier SemWeB annotates Web pages with metadata using a modified GATE framework (Cunningham et al., 2002). We have extended GATE to use linked data in the semantic annotation process. Since IE requires pre-processing, SemWeB uses predefined ontologies, particularly the ECS ontology (ECS Ontology, 2006), DBpedia (DBpedia, 2008) and DBLP (DBLP, 2008). Once a Web page is annotated, from the sidebar users can use the ontological concepts of the ontology to embed hyperlinks in the Web page. When users click on the embedded links, then more information and links are presented based on the metadata of the resource. In addition, in SemWeB we generate implicit links and links to related Web resources by employing goal services. Goal services are shown at the sidebar.

For supplying personalization on different Web domains, we generated a new behaviour-based and ontology-driven user model. In the user model, information about the user's browsing interests, goals, and expertise are represented. Different adaptation is provided, such as adaptive link generation, adaptive text generation and link annotation. This work has been published in (Şah et al., 2008), (Şah et al., 2008b) and (Şah et al., 2009). The proposed Semantic Web browser, SemWeB, is also evaluated using a system-based evaluation and a scenario-based evaluation.

1.6 Thesis Structure

The rest of the thesis is structured as follows:

Chapter 2: Hypertext. This chapter provides a look at the history of hypertext and describes key systems and philosophies in its development.

Chapter 3: Semantic Web. This chapter provides an in-depth look into the Semantic Web, describes its core technologies and standards, shows examples of Semantic Web metadata (i.e. RDF, RDFS and OWL) and discusses the linked data Web.

Chapter 4: Adaptive Hypermedia (AH). This chapter presents a review of AH systems, adaptation metrics, and AH methods and techniques. It also examines pre-Web AH, Web-based AH and Semantic Web-based AH from the author's perspective.

Chapter 5: Related Works. This chapter presents a review of semantic portals, Semantic Web browsers and semantic annotation research, examines the existing research in those fields and shows a selection of state-of-the art related works from the author's perspective. It also discusses the drawbacks of the existing approaches and the motivation behind designing our proposed Semantic Web portal (SEMPort) and Semantic Web browser (SemWeB).

Chapter 6: SEMPort – A Personalized Semantic Portal. This chapter explains our novel semantic portal. In this chapter the technology of the portal, its functionalites (i.e. semantic navigation, ontology-based search, context-based semantic hyperlinks, personalization, content editing/provision) are explained and discussed in detail.

Chapter 7: Evaluation of SEMPort. This chapter discusses the evaluations of SEMPort undertaken. A structured review is used to assess the usability of SEMPort design. The chapter summaries the approach undertaken and shows the results of the experiment. In addition, an empirical study of SEMPort is done and tested on users. This chapter also explains the study, the results obtained and the user's attitudes to our proposed semantic portal.

Chapter 8: SemWeB – A Personalized Semantic Web Browser. This chapter describes our novel Semantic Web browser. SemWeB adds a semantic layer to the ordinary Web browser using linked data and shows ontology-based hyperlinks and adapts information to the individual users. This chapter explains details of SemWeB, such as our semantic annotation mechanism, link generation system, adaptation mechanisms and the proposed new user model ontology and demonstrates the functionalities of SemWeB using the ECS domain.

Chapter 9: Evaluation of SemWeB. This chapter discusses the evaluations undertaken. A system-based evaluation is performed to test the interoperability, adaptability and scalability of SemWeB using the DBpedia and DBLP domains. Demonstrations of SemWeB on different domains are given and its scalability is discussed. In addition, a scenario-based evaluation of the approach is undertaken. We show the benefits of SemWeB in different user scenarios and demonstrate its functionalities.

Chapter 10: Conclusions and Future Work. This chapter summarises our work and presents possible future directions for the work.

1.7 Declaration

This thesis describes the research undertaken by the author. It is all the original work of the author, except where explicitly stated otherwise.

2 Hypertext

In this chapter, we present a short history of hypertext and discuss important figures and systems in its development. Then the most widely accepted hypertext system, the WWW is explained in detail.

2.1 A Short History of Hypertext

The history of hypertext usually begins with an article entitled “As We May Think” (Bush, 1945), written by Vannevar Bush in 1945 and published in the Atlantic Montly. In this article, Vannevar Bush proposes ideas for a photo-electrical-mechanical device called the *Memex*, which stores textual and graphical information, and can make and follow links between documents. The Memex is designed with scientific researchers in mind: Bush argues that many increasing number of papers, books and reports are the time and classical indexing is insufficient to cope with the records. As Vannevar Bush says *“The human mind does not work that way. It operates by association”* (Bush, 1945). He then proposes the Memex where users can build trails to follow articles and articles are connected by simply pressing a button. Trails are named and stored, so that later users can use the trail again. The trails can be thought of as the first conception of hyperlinks.

Twenty years later, the ideas of Vannevar Bush influenced Ted Nelson and Douglas Engelbart. In 1965, Ted Nelson coined the words *Hypertext* and *Hypermedia* in his paper “Complex information processing: a file structure for the complex, the changing and the indeterminate” (Nelson, 1965). He explained hypertext and hypermedia as follows (Nelson, 1965):

"hypertext mean nonsequential writing – text that branches and allows choice to the reader, best read at an interactive screen"

"hypermedia is used as a logical extension of the term hypertext, in which graphics, audio, video, plain text and hyperlinks intertwine to create a generally non-linear medium of information"

In simplest terms, hypertext is non-linear text. Text documents are connected using links. Hypermedia extends the notion of the text in hypertext with graphics, video, audio, animation and other media forms. The items that are linked together are called *nodes*. Links may have type or other attributes and can be uni- or bi-directional, and are often called *hyperlinks*. As a result of hypertext, a network of interconnected nodes is created, which can be navigated by following links.

In the 1960s, Ted Nelson started the *Xanadu* project[5] (Wikipedia, 2008). The idea was to create a universal document database (docuverse) such that documents can be linked from any substring to other documents. Xanadu also aims to maintain versions of the documents and contents to solve the broken link problem. The first implementation (albeit incomplete) was released in 1998 and in 2007, XanaduSpace 1.0 released.

In 1968, Douglas Engelbart demonstrated the first working hypertext system, *oNLine System* (NLS), with a ninety minute live presentation at the Fall Joint Computer Conference in San Francisco. NLS was used for cross-referencing research papers for sharing among geographically distributed researchers (Engelbart, 1963). For this system, Engelbart invented the first graphical user interface and the computer mouse. Engelbart's vision of hypertext mainly focused to human communication and collaboration though the computer.

The second working hypertext system was the *Hypertext Editing System* (HES), which was developed by researchers lead by Andries van Dam and Ted Nelson in 1969 (Carmody et al., 1969). HES was used by NASA to produce the user manuals for the Apollo mission to the Moon. Later, Andries van Dam and his team developed the File Retrieval and Editing System (FRESS), which incorporated some concepts from HES system (Nelson, 1974). In FRESS, the speed of the system was improved, bi-directional

---

[5] http://xanadu.com/ [last accessed, 17/6/2008]

links were implemented and links could be stored in link databases separately from the documents.

Influential projects continued to emerge. In 1975, *ZOG* system was developed (Akscyn et al., 1988). The ZOG database consisted of frames of text and some hypertext for cross-referencing ability. Later, the *Knowledge Management System* (KMS) was developed from the ZOG system. KMS managed both text and graphics on a local area network (Akscyn et al., 1988). In 1978, a team at MIT, lead by Andrew Lippman, implemented the *Aspen Movie Map*, the first true multimedia application, including a videodisk (Lippman, 1980). In 1985, Xerox released *NoteCards*, which allowed the scrolling of windows for each notecard (Halasz, 1987). Hyperties and Guide was the first commercial hypertext products released for PC systems. Hyperties worked on plain text screens of IBM PCs and PS/2s (Schneiderman, 1987). Guide developed by the University of Kent was initially a product for the Macintosh platform by Office Workstations Ltd (Brown, 1987). Later IBM PC version was developed and for sometime Guide was the only hypertext system available for both platforms. In 1987, Apple introduced HyperCard (Goodman, 1987). Apple delivered HyperCard free with every Macintosh and it became the most widely used hypertext system at that time.

One of the first true open systems is the Sun's link service developed by Amy Pearl (Pearl, 1989). Using the link service, hypertext (or hypermedia) could be accessed by an open set of applications in a distributed environment through a communication protocol. Link servers were used to store links and content references separately from the original document. Therefore, links could be added to different media. In 1985, Brown University introduced *Intermedia* (Meyrowitz, 1986). The Intermedia project was a pioneer in the development of open hypermedia systems. The distinctive feature of the Intermedia was the separation of links and document data, where information about links was stored to link databases. Intermedia's aim was to ease link management, such that links could be shared by participating applications. In addition, links were grouped into collections and different set of links were displayed depending on the selected collection. Later, Intermedia's vision is continued by Microcosm.

Other three important hypertext systems in the history of hypertext are Microcosm, WWW and Hyper-G, where all started life in 1989 independent of each other. At the end of 1980s, the open hypermedia system Microcosm was developed by researchers lead by Wendy Hall in the University of Southampton (Fountain et al., 1990, Hall et al., 1996). The design idea of Microcosm was to build an open hypermedia system for linking diverse sources and formats to support hypermedia based on the open hypermedia link service. In the Microcosm, links were also separated from the documents and link data were stored in link databases, which were called *linkbases* to support open hypertext (hypermedia) functionality on diverse document formats (i.e. ASCII text, bitmaps, digital video, etc.). Linkbases were used to add hypertext functionality and Microcosm supports three primitive links types: specific links (a link may be followed from a specific selection from a specific document), local links (a link may be followed at any place in a specific document) and generic links (a link may be followed from wherever the source selection occurs). In 1989, the WWW was invented by Tim Berners-Lee and Robert Cailliau. The WWW is the most widely used and successful hypermedia system to date and is considered a milestone in the history of hypertext systems. It will be explained in more detail in the next section. After the development of the WWW, in 1994 the Distributed Link Service (DLS) was implemented from Microcosm, where links can be incorporated from wide range of network information (Carr et al., 1995). The Hyper-G project started in 1989 at Graz University of Technology by Hermann Maurer and his team. The system intended to improve shortcomings of the WWW, such as hyperlink management, searching, dynamic content, maintenance of large datasets, authoring and scalability. In 1995, they released the first commercial version. However, at the time it was released, the WWW had already become widely used and Hyper-G did not get broad acceptance. For more information about the history of hypertext, the reader is referred to the book of Jakob Nielsen (Nielsen, 1995).

2.2 The World Wide Web and Hypertext

The World Wide Web was born at the International Laboratory for Particle Physics (CERN) in Geneva. CERN has several thousand people, which work all around the world. They need to exchange documents electronically, but geographical differences and different computer system environments were making this a big problem.

To solve these problems, in 1989, Tim Berners-Lee and Robert Cailliau proposed a distributed hypertext system, which they called *World Wide Web* (WWW) (Berners-

Lee and Fischetti, 1999; Cailliau, 1995). They had two main goals in designing the system: *open design* so that the system operate on different computer architectures and *network distribution* so that the system can be shared over distributed communications system. For this purpose, Tim Berners-Lee implemented the first Web browser with an integrated editor (Nexus) for creating hypertext documents. The first Web server became operational at the end of 1990. In a short time, many Web servers rapidly became operational and in 1993, the WWW became public for everyone to use and built on it. In popularizing of the WWW, the Mosaic Web browser[6] played a key role, which was released in 1993. The Mosaic's easy-to-use interface and simple installation contributed to the wide spread use of the Web in the general public. The WWW continues to develop and grow this today. Now the WWW is a global hypertext system that billions of people use everyday for entertainment, communication, business and many other purposes.

In the pre-Web world, documents were the first-order objects and aim was to publish them publicly. Moreover, the technology was lacking to create and publish documents effectively over distributed heterogeneous networks. The WWW provided the technology for people to create documents and make them available to others by adding links.

The success of the WWW relies on its operability on different machines, ease of use and the fact that it is built on open standards. After its wide acceptance, in 1994 Tim Berners-Lee founded the World Wide Web Consortium (W3C), which is the main international standards organization for the WWW. W3C was created to ensure the on going compatibility of the standards of the WWW.

The WWW is based on four core technologies: a universal address system, a network protocol for Web servers, a markup language and a Web browser. The overview of the core WWW technologies will be briefly explained in the following sub-sections.

2.2.1 Uniform Resource Identifier (URI) and Uniform Resource Locator (URL)

---

[6] http://www.ncsa.uiuc.edu/Projects/mosaic.html [last accessed, 16/6/2008]

URLs and URIs were introduced by Tim Berners-Lee in 1990 as a short string representation of a resource that is the target of a hyperlink. URIs are unique concatenated strings for identifying things in the WWW (i.e. http://en.wikipedia.org/wiki/Web). URLs are also uniquely concatenated strings for representing the Web address of a page on a server. The term URL is often used as a synonym for URI. URIs and URLs start with the scheme they are pointing to (http, ftp, mailto, etc.).

2.2.2 Hypertext Transfer Protocol (HTTP)

HTTP is a communication protocol for transferring information on the WWW (Fielding et al. 1999). HTTP provides request and response standards between a client and a server. Clients make HTTP requests to a server using a Web browser or a Web spider or other end-user tool and then the server creates resources (i.e. HTML page, images, etc.) and sends back a status line "HTTP 200 OK" and the message containing the requested document. HTTP is a simple protocol based on the Transmission Control Protocol (TCP) on the Internet. The latest version HTTP 1.1 was released in 1999 (Fielding at al., 1999).

2.2.3 Hypertext Markup Language (HTML)

HTML is markup language with a conforming SGML Document Type Definition (DTD) (W3C 1999). It provides a formal language to describe the structure of documents using special HTML elements, such as links, headlines, lists, tables, images and so on. HTML documents are transferred from a Web server to a Web browser through the HTTP protocol. HTML is intended for human usage, such as people need to render documents, read and understand their content. Machines cannot understand this markup.

2.2.4 Web Browsers

In order to view and use the WWW, in 1991 Tim Berners-Lee implemented the first WWW browser and HTML editor, which is named WorldWideWeb. To save confusion it was later renamed Nexus. At that time, it was the only way to see the Web. But the final link in the chain of Web technology standards was completed by Mosaic. In 1993, Mosaic Web browser is released, which is browser-only software for viewing and navigating the Web, based on the technologies (URL, Gopher, FTP, TCP, HTML, etc.). Mosaic was easy-to-use and played a key role in globally acceptance of the Web and development of future Web browsers. In the following years, other Web browsers emerged. In 2003, safari Web browser[7], in 1994, Netscape Navigator Web browser[8], in 1994, Mozilla Firefox[9], in 1995, Internet Explorer Web browser[10] is released. The Web browsers play a vital role for the growth and the wide acceptance of the WWW.

2.3 Discussion of Hypertext

One topic which needs more attention in hypertext systems is typed links. A typed link is a *"link is to another document or part of a document that includes information about the character of the link"* (Trigg, 1983). With a typed link, different kinds of relationships between documents can be made explicitly. Using the typed links, users can select/search what kind of documents they are looking for by looking at the purpose of the link before navigating to another document. In addition, using typed links, a hypertext system can display certain types of links in a different way. Typed links were a common feature in pre-Web hypertext systems, such as demonstrated by Xanadu and Notecards. With the WWW, typed links has not been supported until HTML version 4.0 (W3C, 1999b), since the lack of standardized link attributes. On the other hand, in the Semantic Web, typed links are the key of the technology where they are utilized to represent different relationships between resources using ontologies. With HTML version 4.0, typed links are also introduced to the WWW (i.e. *rel* attribute for forward relationship). In addition, with the introduction of RDFa (RDF in HTML attributes) (W3C, 2008d), it is possible to define different kinds of relationships between documents/objects in an HTML page using vocabularies of the Semantic Web.

2.4 Chapter Summary

In this chapter, we looked back at the history of hypertext and discussed important figures and the systems. Then, the core technologies used in the current Web, the

[7] http://www.apple.com/safari/ [last accessed, 16/6/2008]

[8] http://browser.netscape.com/ [last accessed, 16/6/2008]

[9] http://www.mozilla.com/en-US/firefox/ [last ccessed, 16/6/2008]

[10] http://www.microsoft.com/windows/products/winfamily/ie/default.mspx [last accessed, 16/6/2008]

WWW, are discussed, such as universal address system, HTTP protocol, HTML markup language and Web browsers. Finally, we discussed the typed link functionality of hypertext systems.

The current Web is intended for humans to share documents, therefore it does not support machines and automated processing. It is a challenging task for users to find the information they are looking for and difficult for applications to share information on the Web. Alternatively, a machine processable Web could link data instead of documents and data could be shared by communities, processed automatically, and help to support users with their everyday activities on the Web. This new Web, the Semantic Web, is an extension to the current Web and provides machine processable semantics to the Web content. The aim is to overcome the shortcomings of the current Web in automated processing, information discovery, and interoperability as well as the reuse of data between applications. The next chapter explains the Semantic Web in detail.

3 The Semantic Web

The Semantic Web is derived from Tim Berners-Lee's vision of the Web as a universal medium for data, information and knowledge exchange. Its creator, Tim Berners-Lee defines the Semantic Web as (Berners-Lee at al., 2001):

> *"... an extension of the current Web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation"*

The current Web is intended for the use of humans rather than machines: the WWW technology helps people to publish and share *documents* on the Web and machines cannot interpret this data. Alternatively, the Semantic Web represents data about data (metadata) and it is processable by machines. For instance, information about Web resources is explained using formal languages.

The Semantic Web encourages people to publish and share their *data* and add links to other data. The vision of the Semantic Web is "*an extension of Web principles from documents to data*" (Berners-Lee et al., 2006). Berners-Lee et al., defines Semantic Web as "*a technology for sharing data, just as the hypertext Web is for sharing documents*" (Berners-Lee et al., 2006b). As a result of this, data can be shared by diverse communities, processed automatically by tools, interoperable across applications and inferenced to find implicit knowledge. In addition, data can be used to enhance information discovery, so that search results can be improved.

The Semantic Web is an engineering solution that provides a common framework for creating, publishing and linking *data* in machine processable form. To achieve this, The Semantic Web approach develops languages, methods and tools for expressing and accessing information in a machine processable form.

3.1 The Semantic Web Technologies and Standards

To support the vision of a Web of linked *data*, the Semantic Web identifies a set of technologies, tools and standards. The layered architecture of the Semantic Web is outlined by Tim Berners-Lee (Berners-Lee et al., 2006; Berners-Lee, 2000) in Figure 2-1, the so-called "Semantic Web Stack".

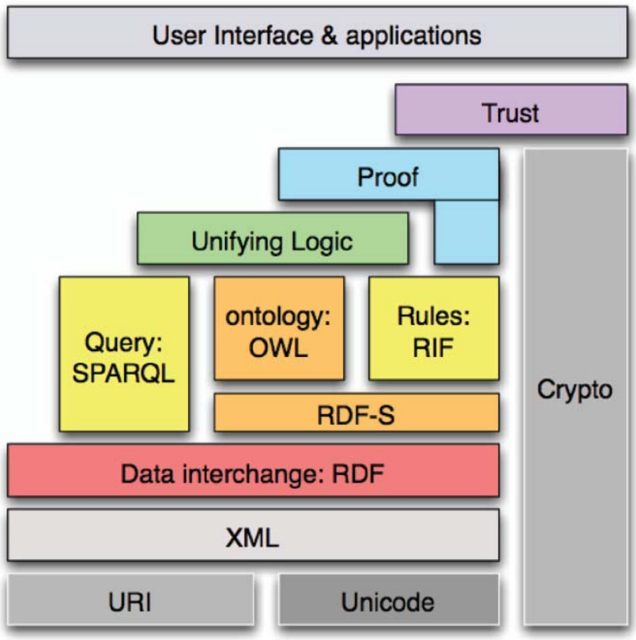


Figure 2-1 The Semantic Web Stack (Berners-Lee et al., 2006)

The first layer of the Semantic Web layer cake is Unicode and URI. These are the foundations of the stack. They are used to identify resources with unique identifiers. The second layer is XML and XML Schema, which are syntax languages for representing structured information. The third layer is RDF, which is more expressive than XML and the data model for the Semantic Web. The next layer is RDF Schema (RDFS), a vocabulary language for RDF. In the next layer, the OWL ontology language and the RIF rule language for the Semantic Web are presented. SPARQL is a query language and protocol for the Semantic Web. On top of the representation layers is the Unifying Logic layer, which is used to reason over RDF statements. The next layer is the Proof which is used to validate RDF model. The trust layer is the next layer to support the security of the Semantic Web. Finally, the user interface and applications layer sits on top of the Semantic Web stack. An overview of the Semantic Web technologies is presented in the following sub-sections.

3.1.1 Unicode and Uniform Resource Identifier (URI)

Unicode is the standard for computer character representation and URI is a string of characters used to uniquely identify resources on the Internet. They are the foundations of the Semantic Web for identifying resources with a concrete serialization syntax.

3.1.2 Extensible Markup Language (XML)

XML is a simple, flexible text format derived from SGML (ISO 8879) (W3C, 2006). XML is a W3C recommendation and was developed to facilitate the sharing of structured information between various applications on the Web. It is also used to encode documents and serialize data. XML is a markup language and took its "extensible language" name because tags are not predefined. People can define their own tags, since XML is designed to be self-descriptive.

An XML document consists of three parts (Figure 2-2): prolog, entity and attribute. The prolog part is the first line and appears before the root element. It contains the XML declaration and reference to other documents (<?xml version="1.0"?>). Entities represent things in the document (e.g. Book). Finally, an attribute is a value inside the opening tag of an entity (e.g. ID).

```
<?xml version="1.0" encoding="UTF-8"?>
<Book>
   <Authors ID="123456">
      Erik T. Ray
   </Authors>
   <title>
      Learning XML
   </title>
</Book>
```

Figure 2-2 An XML document example

XML documents should be well-formatted (hence valid), ensuring that all XML-aware software can read and understand the relative arrangement of information within them. This is done by applying some syntactic rules. XML structure can be defined using Document Type Definition (DTD), XML Schema (W3C, 2004d) or RELAX NG.

3.1.3 Resource Description Framework (RDF)

RDF is a metadata data model for making statements about Web resources in the form of subject-predicate-object expressions (triples) (W3C, 2004). The subject denotes the resource, the predicate describes aspects of the resource or creates relationships between other resources (i.e. relationships to objects). All resources (subjects, predicates, objects) are identified by unique Web identifiers (URIs). This mechanism of describing resources enables automated storage, sharing and machine readable data on the Web.

RDF is an abstract model and can be serialized and presented in different formats, such as RDF/XML syntax (Figure 2-3), Notation 3 (N3) (Figure 2-4), directed labelled graph (Figure 2-5), etc.

```
<rdf:Description rdf:ID="http://books.com/book1">
   <dc:title>
      A Semantic Web Guide
   </dc:title>
</rdf:Description>
```

Figure 2-3 RDF/XML serialization

```
<http://books.com/book1> <dc:title> "A Semantic Web Guide".
```

Figure 2-4 N3 serialization

Figure 2-5 RDF as directed labelled graph

In the RDF model, two types of triples are found: *Literal triples* and *RDF links*. Literal triples are used to describe properties of a resource in the form of *"(resource, resource, literal)"* pattern. For instance, a book's title is represented by literal triples (see Figure 2-5). On the other hand, RDF links represent typed links between two resources in the form of the *"(resource, resource, resource)"* triple pattern (see Figure 2-6).

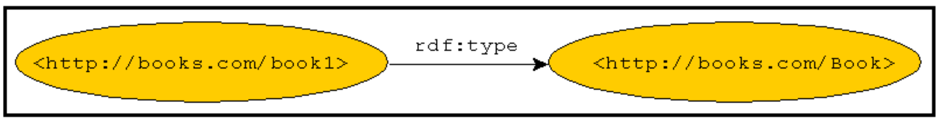


Figure 2-6 Representation of an RDF link as directed labelled graph

RDF is a general purpose language for representing information on the Web and does not include information about vocabularies. RDF may need application specific classes or properties conformed by vocabularies. Therefore, a schema language is needed to define a predefined vocabulary used at RDF metadata.

3.1.4 RDF Schema (RDFS)

RDFS is the vocabulary language for RDF (W3C, 2004b). It is a framework for describing application specific classes and properties. For instance, RDFS models a domain in a hierarchical fashion using *"rdfs:subClassOf"* relationship (Figure 2-7). However, in a vocabulary, more complex relationships may exist between classes and RDFS is not capable of representing this. More expressive languages are needed such as OWL.

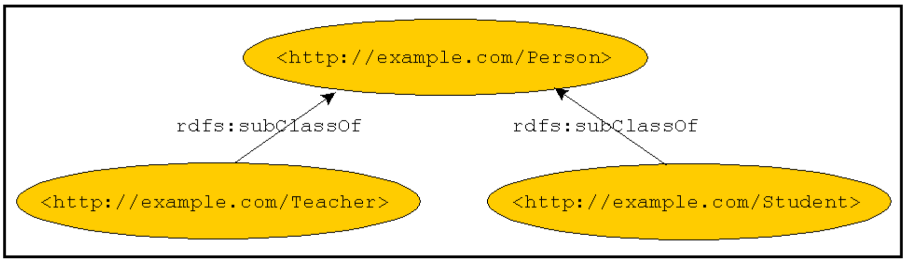


Figure 2-7 RDFS as directed labelled graph

3.1.5 Web Ontology Language (OWL)

The term ontology originates from philosophy, which means the study of the nature of existence. In computer science, ontology has a different meaning, where Gruber defines it as *"an explicit and formal specialization of a conceptualization"* (Gruber, 1993). In the Semantic Web, ontologies describe a set of concepts and relationships between them in a machine processable form.

The W3C recommended ontology language for the Semantic Web is OWL (W3C, 2004c). OWL has its roots in DAML+OIL. DAML+OIL was first developed as an Agent Markup Language from the need for a powerful ontology language. The W3C ontology working group subsequently revised DAML+OIL and developed OWL. OWL allows ontologies to be referred in other ontologies; therefore ontologies can be used in a distributed fashion. OWL is more expressive than RDFS (i.e. disjointness of classes). OWL can also be used to express different relationships between resources on the Web, such as supervisorOf in Figure 2-8.

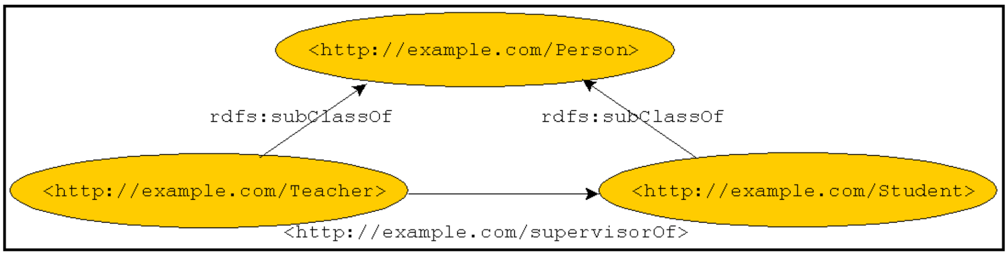


Figure 2-8 OWL as a directed labelled graph

OWL has three sub-languages with different levels of expressiveness and reasoning.

- **OWL Lite:** provides classification hierarchy and simple constraints (i.e. cardinality constraints only accept values 0 or 1). Thesauri and taxonomies are examples to OWL Lite.
- **OWL Description Logic (DL):** supports all language constructs of OWL and supplies the maximum expressiveness while maintaining computational completeness (all conclusions are completed), finite computation time and providing reasoning algorithms.
- **OWL Full:** includes all OWL language constructs, provides maximum expressiveness and freedom of RDF with no computational guarantees.

3.1.6 Rules Layer: Rule Interchange Format (RIF)

The aim of the rules layer is to provide appropriate languages for representing rules on the Semantic Web and currently it sits along side the ontology layer. The RIF[11] is a W3C working draft recommendation, which aims to develop an interchange format for different rule languages and inference engines, so that machines can share rules on the Semantic Web (W3C, 2008c).

3.1.7 SPARQL Query Language for RDF

SPARQL has recently become a W3C recommended query language (W3C, 2008) and protocol (W3C, 2008b) for RDF.

As a query language, SPARQL is a syntactically SQL-like language for querying RDF graphs using pattern matching, such as conjunctive patterns, value filters, optional patterns, and pattern disjunction. SPARQL queries can be formed in four different

[11] http://www.w3.org/2005/rules/wiki/RIF_Working_Group [last accessed, 17/6/2008]

ways: *SELECT, CONSTRUCT, ASK* and *DESCRIBE*. Select queries return variables and their bindings directly. The results can be accessed by the query engine's API or can be serialized into XML or RDF. Construct queries form an RDF graph specified by the patterns defined in the query. The output RDF graph is generated based on the form specified in the construct query. Ask queries are used to check if a query pattern exists or not. Therefore, the result is binary (yes or no). Describe queries are used to form an RDF graph about resources identified by the query. All available information about resources is given in the RDF graph.

As a protocol, SPARQL provides a simple interface via HTTP or SOAP, so that clients remotely invocate SPARQL queries to an endpoint.

3.1.8 Logic Layer and Inference

The logic layer supports formal languages for making inferencing on the Semantic Web. The aim is to find implicit knowledge and to uncover inconsistencies in the metadata using semantic reasoners. Reasoners are software tools for inferring conclusions from asserted facts. Most of the semantic reasoners utilize first-order predicate logic for performing inferencing; reasoning is based on inference rules, which are generally specified according to the ontology language. Jena[12], Pellet[13], KAON2[14], and FACT[15] are examples of semantic reasoners.

By using the logic layer, automated reasoners can deduce conclusions from the given knowledge. This can be illustrated with a software agent example. Software agents gather information on the Web, compare information with user choices and make decisions. Such a logic layer can support decidability on the Web. Currently there is not a standard W3C language for this layer, however the W3C recommended draft for RIF supports interoperability between different rule languages (W3C, 2008c).

---

[12] http://jena.sourceforge.net/ [last accessed, 17/6/2008]

[13] http://pellet.owldl.com/ [last accessed, 17/6/2008]

[14] http://kaon2.semanticweb.org/ [last accessed. 17/6/2008]

[15] http://www.cs.man.ac.uk/~horrocks/FaCT/ [last accessed, 17/6/2008]

3.1.9 Proof Layer

The aim of the proof layer is to validate information generated as RDF, such as the provenance knowledge or the form of reasoning that is used. At this stage of the development of the Semantic Web, this problem is not yet been resolved.

3.1.10 Trust Layer

The main point of the Web is "anyone can say anything about anything". Therefore, when we are selecting a resource on the Web we are putting our trust in it. We make trust judgments based on a source's perceived reputation or previous personal experience and so on. The same is true for the Semantic Web. Encryption mechanisms should allow people to sign up to trusted metadata on the Semantic Web. In addition, semantic agents need to make judgements when alternative sources of information are available. The aim of the trust layer is to shed light on these problems.

3.1.11 User Interface and Applications

Semantic Web technologies are basically machine-oriented: formal models are used to express data so that machines can reason on them. However, Semantic Web applications are not only machine-oriented, they will also support users. This layer of the Semantic Web stack is for user-oriented applications to improve the user's experience on the Semantic Web. Examples of user-oriented Semantic Web-enabled interfaces to support user access to the Semantic Web are as MSpace (Schraefel et al., 2005) or interfaces like COHSE (Carr et al., 2001) and Magpie (Dzbor et al., 2003).

3.2 Semantic Web as a Web of Linked Data

The Semantic Web technologies provide an environment to create and publish structured data on the Web. According to Tim Berners-Lee, the metadata could be more useful, if it is represented with common vocabularies (reusing exiting ontologies) and interconnected to different datasets on the Web (links between datasets) (Berners-Lee, 2006c). From these needs, the term linked data has been introduced by Tim Bernes-Lee in his Linked Data Web architecture note (Berners-Lee, 2006c). The term refers to exposing, sharing and interlinking structured data on the Semantic Web. The rationale behind it is that the value and usefulness of data increases the more it is connected to other data. Therefore, it is about making links. This can be illustrated with an example. I want to search for *"all publications from Semantic Web related conferences in 2007"*. Although such information is available on the Web, either it is not represented in RDF format or represented in RDF format but disconnected from related resources. When data is published on the Semantic Web and connected to other datasets, information discovery can be improved.

Linked data is an outcome of a community effort. The W3C *Semantic Web Education and Outreach* group's *Linking Open Data Community Project* aims to increase the Web of linked data by publishing various open datasets as RDF on the Web and by connecting them to different data sources. Figure 2-9 shows the extent of published linked data in September 2008.

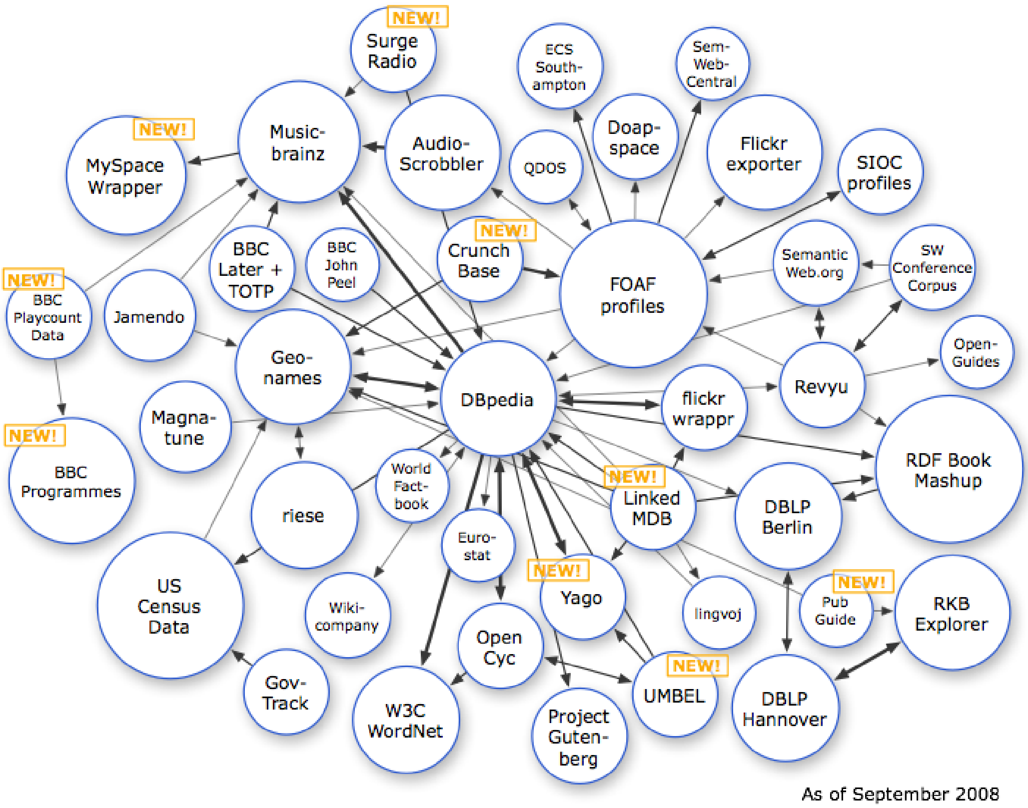


Figure 2-9 Open linked data on the Web, September 2008[16]

Some examples of linked data are: the DBpedia (DBpedia, 2008), the DBLP (DBLP, 2008), Geonames (Geonames, 2008) and ECS Southampton (ECS Southampton, 2006). DBpedia is a community effort that extracts structured information from the Wikipedia

---

[16] http://www4.wiwiss.fu-berlin.de/bizer/pub/lod-datasets_2008-09-18.html [last accessed, 22/2/2009]

and publishes this information on the Semantic Web and interlink these resources to various datasets (Auer et al., 2007). The DBLP Bibliography database provides metadata about scientific papers, conferences, journals and authors. Geonames provides metadata about geographical data (i.e. places names in different languages, population, etc.) and latitude/longitude coordinates of places. The ECS Website publishes metadata about people, publications, modules, etc. within the School of Electronics and Computer Science in the University of Southampton.

3.2.1 Basic Principles of Linked Data

Tim Berners-Lee outlined four principles of Linked Data (Berners-Lee, 2006c):

- Use URIs to identify things that you expose on the Web as resources.
- Use HTTP URIs, meaning that an application can look up a URI over HTTP protocol and retrieve RDF data about the identified resource.
- Provide useful information about the resource when its URI is dereferenced.
- Interlinked with other data. The resource description should contain links to related URIs within RDF statements or as *rdfs:seeAlso* links or *owl:SameAs* links.

Linked data is published on the Web by using RDF. RDF instance data can be serialized in a number of ways, formats including RDFa (RDF in HTML attributes), XML, Notation 3 (N3), Turtle (terse RDF triple language) and others.

RDF links are the basis of the linked data, for instance they allow us to navigate on the Web of linked data from one resource to another. RDF Links can interlink resources from different domains. These are known as *external RDF links*. External RDF links have subject URI and object URI from different data sources (e.g. foaf:knows in Figure 2-10). Using external RDF links, different RDF graphs on the Web can be merged together.

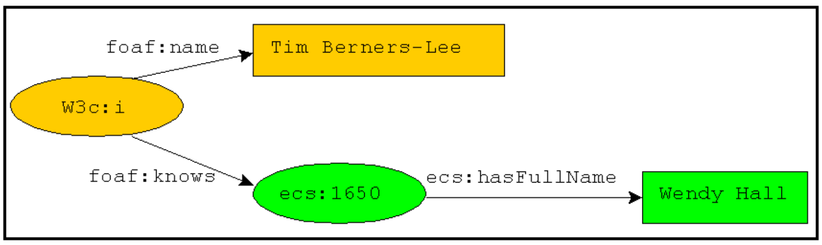


Figure 2-10 An example of an external RDF link

3.2.2 URI Dereferencing

URI dereferencing means looking up a URI on the Web using HTTP protocol and retrieving information about the resource (Bizer et al., 2007). The W3C TAG draft makes a distinction on how to deference a URI when it identifies an information resource and non-information resource (W3C, 2007). Traditional documents on the Web, which have human-readable presentations, are accepted as information resources, such as pages, images and media files. Non-information resources are resources that do not have an obvious presentation, such as ontologies and things (Person, Place, etc.).

HTTP URIs can be dereferenced in different ways. First, when a URI identifing an information resource is requested from the server, the server creates a human readable representation and sends it back to client using the HTTP response code *"200 OK"* (dereferenced directly). Second, when a URI identifing a non-information resource is requested from the server, currently people use a Web architecture trick to dereference a resource. Instead of sending the representation of the resource, the server returns the URI of an information resource, which describes the non-information resource using the HTTP response code *"303 See Other"*, which is also called *303 redirect*. The client then dereferences the new URI of the information resource and obtains the representation describing the non-information resource.

Alternatively, providers of the non-information resources can use two different approaches for serving information resources (W3C, 2007b): *Hash URIs* and *303 redirect*. The first solution is to use Hash URIs for non-information resources, where the URI contain a fragment separated by "#" symbol (i.e. http://www.example.com/about#bob). Before the client requests this URI, the "#" part is stripped off, resulting in "http://www.example.com/about". This new URI serves as an RDF document containing the RDF description of all the resources identified using the same URI domain.The second solution is 303 redirect, which is also known as *content negotiation*. In content negotiation, the client sends the HTTP headers indicating what kinds of presentation they prefer. For example, for retrieving HTML documents, "accept: text/html" header or for retrieving RDF description of a resource "accept:application/rdf+xml" header is sent together with the requested URI. Then, the server redirects to an information resource based on the client's preference (illustrated in Figure 2-11). Therefore, data sources provide three different URIs for non-information resources. For instance, in ECS Southampton, three different URIs are used to serve information about Wendy Hall as shown in figure 2-12.

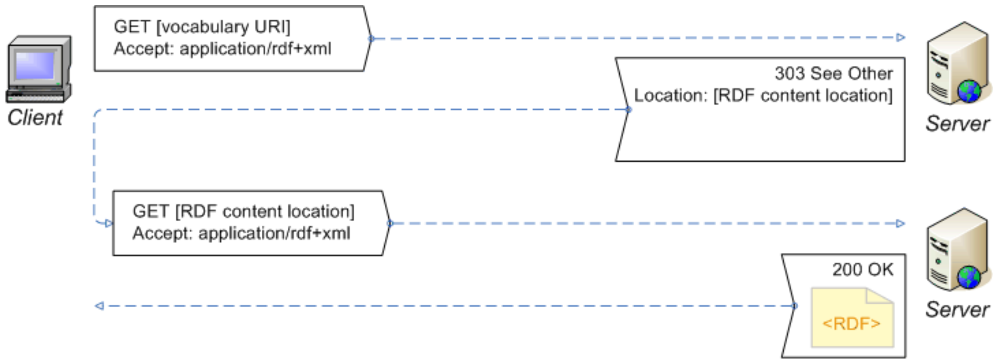


Figure 2-11 HTTP content negotiation (Bizer et al., 2007)

```
http://id.ecs.soton.ac.uk/person/1650
      URI for Wendy Hall, the person
http://www.ecs.soton.ac.uk/people/wh
      HTML page about Wendy Hall
http://rdf.ecs.soton.ac.uk/person/1650
      RDF description about Wendy Hall
```

Figure 2-12 URIs used in ECS Southampton

3.2.3 Examples of Ontologies and Vocabularies for Publishing Linked Data

A number ontologies and vocabularies are commonly used for publishing metadata, such as Dublin Core – for cross-domain information resource description (Dublin Core, 2006), Simple Knowledge Organisation Systems (SKOS) – for representation of thesauri, classification schemes, taxonomies or other type of structured controlled vocabularies (SKOS, 2008), Friend of a Friend (FOAF) – for describing people, their activities and their relations to other people and objects (FOAF, 2007), Description of a Project (DOAP) – for describing open source projects (DOAP, 2008), Semantically-Interlinked Online Communities Project (SIOC) – for expressing the data contained both explicitly and implicitly in Internet discussion methods (SIOC, 2008), and so on.

3.2.4 Serving Linked Data

Linked data can be created from databases using a D2R server (a tool for publishing relational databases on the Semantic Web), or from social networking Web sites using mashups or created manually using RDF files.

A D2R server is a tool, which maps the contents of databases to RDF (Bizer and Cyganiak, 2007). Based on this mapping, a D2R server allows browsing and searching of the RDF representation of the database by assigning dereferenceable URIs. The RDF description of a resource can be accessed from a D2R server using the dereferencing interface, which supports content negotiation and serves RDF and XHTML representation of resources. Additionally, a SPARQL interface (SPARQL endpoint) is provided, from which applications can query the database using the SPARQL query language over the SPARQL protocol. For instance, the DBPedia (DBpedia, 2008) and DBLP (DBLP, 2008) databases use D2R servers for publishing RDF content.

In addition to public databases, there are major data sources published by third parties using Web APIs (i.e. Amazon, Yahoo, Google and eBay). Mashups like RDF Book Mashup (Bizer et al., 2007b) aim to combine data from multiple sources and publish them with dereferenceable URIs on the Semantic Web.

Alternatively, RDF files can be created manually, such as FOAF files. When serving a static RDF file at, say, http://example.com/foaf.rdf, the URIs for non-information resources should be unique in that file and served appending fragments to file name using hash URIs (i.e. http://example.com/foaf.rdf#me).

3.3 Chapter Summary

In this chapter, Semantic Web standards and technologies and the concept of the linked data Web are discussed. We can say that the Semantic Web is on the verge of take off. The development of the Semantic Web is continuing and the majority of the necessary standards (i.e. RDF, RDFS, OWL, etc.) and languages (i.e. SPARQL, RIF) have been developed. In addition, people from different communities have started to realize the benefits and advantages of the Semantic Web and take part in its development. The evolution of the linked data is a proof of it. Useful metadata about different domains are now available on the Semantic Web and this metadata can be used for enhanced information discovery, sharing, reasoning and interoperable systems. The Semantic Web is a Web of data rather than just documents. We can use this data for automated information discovery, reasoning, searching and reuse it to generate even more powerful applications. In the next chapter, Adaptive Hypermedia (AH) research will be discussed.

4 Adaptive Hypermedia

AH is a direction of research within the areas of hypertext (hypermedia) and user modeling. This research aims to increase the functionality of a hypermedia system by tailoring it to the individual user.

AH systems employ a user model to store the goals, preferences, and knowledge of individual users and apply this model throughout the interaction with the user, in order to adapt to their needs (Brusilovsky, 1996). AH systems are useful in any application where the system is used by people with different goals and knowledge and where the hyperspace is reasonably big. Different users may be interested in different information and may use different links. AH tries to assist the user to navigate by adapting the contents displayed by using the knowledge represented in the user model. This is done by using AH technologies.

In this section, we briefly discuss mechanisms used to adapt AH systems (AHSs), AH methods and techniques, application areas of AH systems, pre-Web AH, Web-based AH and Semantic Web-based AH.

4.1 Adaptation Mechanisms

Most of the AHSs use different mechanisms for adaptation. Generally, three different kinds of data can be utilized: user characteristics, user's individual traits and the user's environment.

4.1.1 User Characteristics

Different user features are employed in AHSs: knowledge, goal, background, experience, preferences and interests.

One of the most popular user characteristic is knowledge. The user's knowledge is mostly represented as *overlay model*. The overlay model represents the user's knowledge as a "concept-value" pair. For each domain concept, the overlay model stores the user's knowledge level (Eklund et al., 1997). The user's knowledge is mostly used in educational hypermedia applications, such as in InterBook (Eklund et al., 1997), ELM-ART (Weber and Brusilovsky, 2001), AHA (De Bra et al., 2006) and AHAM (De Bra et al., 2002).

The other characteristic used in AHSs is the user's goal/task. Determining the user's goal is a hard task and as it can often change from session to session. Goals can be inferred by the system or explicitly entered by the user, and such adaptive systems utilize these goals for the adaptation as represented by (Speretta and Gauch, 2005).

The user's previous experiences are often used in AHSs, such as background information or experience. Experience mainly contains information about the familiarity of the user with the topic or with the underlying hyperspace system. This information is often represented by *stereotypes* (Kobsa, 2001). Stereotypes model specific groups of users that have common characteristics.

User preferences are also widely used in AHSs. Different users may prefer some links or information over others. This information can be entered explicitly by the user or inferred implicitly by the system. Preference information is very useful in Information Retrieval (IR) systems, where users explicitly or implicitly show preferences to the system. MyYahoo (Manber et al., 2000) and iGoogle (iGoogle, 2008) use user preferences to present personalized contents (i.e. customization).

User interests are becoming popular with the emergence of adaptive IR systems. User interests can be modelled as short-term interests or long-term interests and can be utilized for adaptation.

4.1.2 *User's Individual Traits*

The user's background, interests, goals, etc. change frequently. On the other hand, the user's individual traits (i.e. personality factors) do not change at all or only over a long period of time. Examples of individual traits are cognitive factors and learning styles. Individual traits can be obtained by interviews using especially designed psychological tests. For instance, iClass (Turker et al., 200), APeLS (Conlan et al., 2002) and (Bajraktarevic et al., 2003) incorporates learning styles for adaptation.

4.1.3 *Environmental Data*

Environmental data makes use of the user's location and platform for adaptation. Since different users can access the same information from different media platforms (i.e. mobile phones, Personal Digital Assistants (PDAs), etc.), information is adaptively shown. These approaches try to overcome platform limitations (hardware, software, network bandwidth) for the benefit of the user.

4.2 Adaptive Hypermedia Methods and Techniques

In AHSs, the information space is adapted to different users using AH methods and techniques. Brusilovsky (Brusilovsky, 2001) divides these techniques into two essential groups: *adaptive presentation* and *adaptive navigation support*. An updated taxonomy of AH technologies is presented in Figure 4-1 (Bailey et al., 2002).

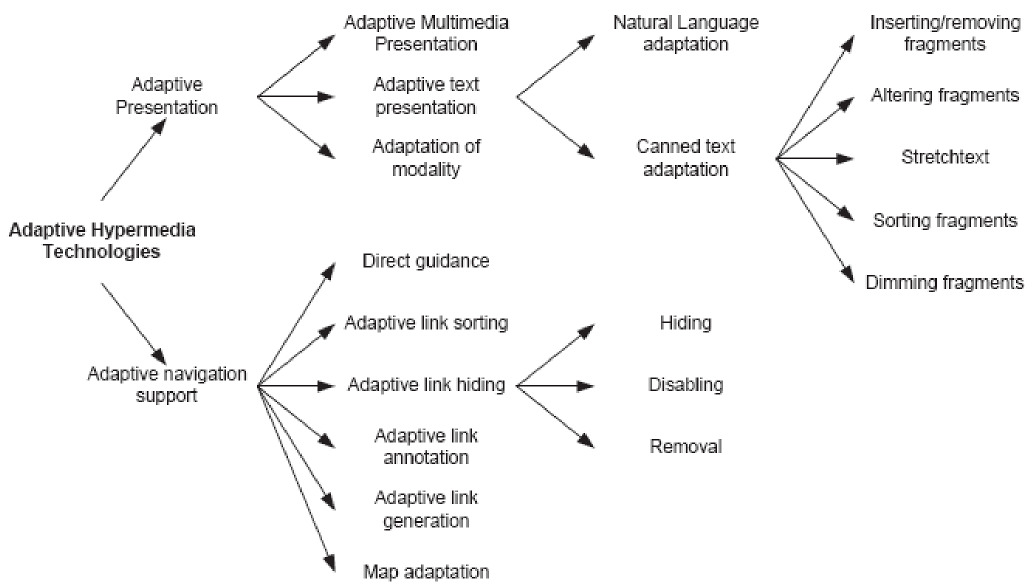


Figure 4-1 Taxonomy of AH technologies (Bailey et al., 2002)

4.2.1 Adaptive Presentation

Adaptive presentation is often performed as a manipulation of text fragments (De Bra et al., 1999). Adaptive presentation techniques provide prerequisites, comparative or additional explanations, give alternative data (present information in different ways), remove or dim information fragments and sort the information according to the user model. For example, an expert user can be provided with extra detailed information or, a novice user can be provided with more explanation.

4.2.2 Adaptive Navigation Support

Adaptive navigation support (Brusilovsky, 2004) focuses on aspects of navigational links, such that the links are adapted based on the user model. Different techniques used for link adaptation include: *direct guidance, adaptive link sorting, adaptive link hiding, adaptive link annotation, adaptive link generation* and *map adaptation*.

- Direct guidance is a technique, which decides the best link for the user to visit according to the user's goal and other information represented in the user model.
- Adaptive link sorting orders all the links in a page according to the user model.
- Adaptive link hiding hides, disables or removes non-relevant links from the page to reduce overload.
- Adaptive link annotation adds various visual or textual clues to the links to help the user select the most relevant one. For instance, the traffic light metaphor for highlighting pages (e.g. green for "ready to read", red for "not ready to read", or yellow for "recommended for reading"). Additionally, different colours and icons can be used to represent the state of the link.
- Adaptive link generation is used for discovering useful links between documents and adding them permanently to the existing set of links. Links are generated based on similarity between elements and dynamically used for recommendations of relevant links.
- Map adaptation adapts the structure of hypermedia maps to the individual users.

4.3 Application Areas of Adaptive Hypermedia

Brusilovsky reviewed AH applications and identified six different areas (Brusilovsky, 1996): *educational hypermedia, online information systems, online help systems, information retrieval hypermedia, institutional hypermedia* and *systems for managing personalized views in information spaces*. Educational hypermedia and online information systems are the most popular application areas of AH.

Educational hypermedia applications try to adapt information to different students using their knowledge of the subject. The goal is to help students to learn the material by showing the appropriate information and hyperlinks based on user's knowledge. A good example for educational hypermedia is Web-based distance education courses such as InterBook (Eklund et al., 1997), ELM-ART (Weber and Brusilovsky, 2001), AHA (De Bra et al., 2006) and AHAM (De Bra et al., 2002).

In the area of online information systems, different kinds of systems are developed, such as *electronic encyclopedias*, *virtual museums* and *e-commerce systems*. Online information systems may have small or reasonably big hyperspaces. The aim is to help users find relevant data using their knowledge level, background and goals.

In online help systems, the information space is much smaller compared to online information systems. The aim of online help systems is to determine the goal of the user and provide the most relevant data based on that.

Information retrieval hypermedia systems are the most challenging in the context of retrieval activity, since they are using the whole Web hyperspace. These systems can be divided into search-oriented systems and browse-oriented systems. Search-oriented systems adapt search results by applying different AH technologies, such as link removal or link annotation to provide users with relevant hyperlinks (Sugiyama et al., 2004; Speretta and Gauch, 2005). The intent of browse-oriented systems is to support navigation using AH technologies, for example the best links are marked using adaptive guidance. Link annotation and link recommendation can also be done based on the user model as discussed in (Yudelson and Brusilovsky, 2005).

Institutional hypermedia systems are developed for employees and to provide access to intuitional hyperspace. They provide personalized access to work area information.

The WWW provides huge amounts of information. Users may have difficulty in finding the information they want. Systems for managing personalized views in information spaces aim to solve this problem by showing a subset of data based on the user's goals and interests. iGoogle (iGoogle, 2008) and myYahoo (Manber et al., 2000) are two examples that provide personalized views. In (Farzan et al., 2007), authors provide adaptive social support during searching and browsing in information spaces.

4.4 Pre-Web Adaptive Hypermedia

The work in pre-Web AH generally studied in closed worlds, so that the underlying document space was known to the authors of the AHS at the time they designed the system. Therefore, changes to the document space are very difficult; a change to a document requires the reorganization of the document space (or at least some of the documents in the document space). In pre-Web AH, the majority of work was on intelligent tutoring systems. A review of work in pre-Web hypermedia can be found in (Brusilovsky, 1996).

4.5 Web-Based Adaptive Hypermedia

With the growth of the WWW, there was a rapid increase in Web-based AHSs. The majority of work on Web-based AH has focused on educational hypermedia, online information systems (electronic encyclopaedia, online help systems, virtual museums, e-commerce, etc.) and information retrieval with personalized views (systems with personalized views) (Brusilovsky, 1996). The latter is the main focus of our research. Our aim is to improve navigation for users by providing personalized data. This can be done by providing hyperlinks to most relevant information items in a page, annotating relevant links with visual cues to help the user select links, and by suggesting information based on the user's interests and browsing activity.

Many approaches have developed to support AH on the WWW, such as ELM-ART (Weber and Brusilovsky, 2001) and InterBook (Eklund et al., 1997). Although these systems support AH on the WWW, they were also able to work on a closed corpus of documents; adaptation worked on documents known to the system.

To open up the limited availability of hypermedia systems, approaches such as the so-called *open-corpus hypermedia systems* have been studied. An open-corpus AH system is an *"adaptive hypermedia system which operates on an open-corpus of documents, e.g., a set of documents that is not known at design time and, moreover, can constantly change and expand"* (Brusilovsky and Henze, 2007; Brusilovsky, 2008). Examples of open-corpus hypermedia systems are Microcosm (Hall et al., 1996), Chimera (Anderson et al., 2000) and Dexter (Gronbaek et al., 1997). These systems allow links and annotations to be added to documents outside the author's control, and are designed to be integrated with any number of applications to provide hypertext functionality to everything from spreadsheets to graphics editors. These systems can also be used by AHSs. For instance, Microcosm provides a framework for building AHSs. An example of the use of Microcosm for educational AH application using static user models is given in (Hothi and Hall, 1998). Some examples of Web-based open-corpus AHSs are KBS hyperbook (Henze and Nejdl, 2001), SIGUE (Carmona et al., 2002) and Knowledge Sea (Brusilovsky and Rizzo, 2003). KBS hyperbook uses indexing to adapt and intergrate information from arbitrary sources in the Web. For instance, hypertext materials are manually structured and indexed with conceptual models by utilizing an object-oriented modelling language. SIGUE is an authoring tool that converts non-adaptive course materials to adaptive material by manually associating domain model concepts to the contents. In the Knowledge Sea project, keyword-based automatic page analysis and self-organazing maps are used to structure Web resources automatically for personalization.

Another approach to open corpus AH is discussed in (Bailey et al., 2002), which describes AH techniques in open hypermedia by relating fundamental open hypermedia model concepts to AH techniques. The work presented in (Bailey et al., 2002) uses more general descriptions of the data objects. Instead of using specific kinds of data object, RDF metadata can be used. RDF annotations provide several possibilities for specifying relationships and association, instead of using special kinds of data object. In addition, the use of the Object-Oriented Hypermedia Design Method for providing personalized links, content, structure, and context in Web applications is described in (Rossi et al., 2001). However, it is not a generic framework.

From the adaptation point of view, in the closed-corpus AHS all the documents and the relationships between them are known at the design time and it is easier for authors to augment adaptation algorithms for delivering adaptation to the users. However, in an open-corpus AHS adaptation is difficult to handle because the documents and their relations are not known at design-time, and the document-space is even expanding. In addition, user models need to be related to the new set of information. For these reasons, the development of open-corpus AHS is very challenging. Our proposed is that the use of the Semantic Web technologies can help solve this problem.

4.6 Semantic Web-Based Adaptive Hypermedia

AH is a research area that can utilize Semantic Web technologies in an attempt to solve some of the problems that AH technologies have with interoperability and reusability; the Semantic Web provides a common language for representing the document space and the user model within a domain. (Kay and Lum, 2003) have already pointed out the advantage of the use of ontologies in user modeling: an agreed ontology and presentations are very important for the employment of the user model by different applications. Documents and user models represented with Semantic Web standards (e.g. RDF) can be used to provide personalization in a broader context, compared to closed architectures with proprietary formats as discussed by (Dolog et al., 2003). In addition, user models that are represented with RDF can have attributes from different standards and can be easily related to different standards as argued by (Dolog and Nejdl, 2003). So far the application area mainly studied using the Semantic Web is that of educational hypermedia.

Semantic Web metadata can be used to provide open-corpus AH and the following papers discuss Semantic Web-based open-corpus AH. (Dolog et al., 2003) discusses a personalization technique on the Semantic Web using a rule-based reasoner using RDF. In this approach, document and user data is annotated with RDF and rules are fired on these RDF models to provide personalization in a learning scenario. In (Dolog et al., 2003b), an RDF-based peer-to-peer network (Edutella) is shown for personalization in open environment of the Web using RDF metadata. (Henze and Nejdl, 2002) tries to apply AH strategies to open-corpus hypermedia in a learning objects repository using the Edutella framework. In this approach, a common ontology (Learning Object Metadata (LOM)) is used to represent learning objects and the user's knowledge is estimated by calculating subgraphs of the ontology with respect to the user's knowledge. To facilitate reusable learning resourses and providing adaptive courses in open learning environments, a multi-model architecture is proposed (APeLS) (Conlan, et al., 2002; Conlan, et al., 2002b; Dagger et al., 2003). The idea is to increase reusability, accessibility and interoperability of learning resources by representing them with standardized markup (i.e. LOM, IMS LIP) and grouping learning resources with similar goals, objectives and learning styles. In (Lawless and Wade, 2006), an architecture is proposed to dynamically harvesting and delivering sources to adaptive elearning systems in an open-corpus content. In this approach, a Web crawler harvests and searches metadata from the repositories and the WWW. Then the metadata is cached and stored by mapping to a fixed ontology for later use by adaptive elearning systems.

4.6.1 AH and Metadata

There have been some efforts to standardize the information about a user which should be maintained by a system. The IEEE Public and Private Information (IEEE PAPI) (IEEE PAPI, 2008), the IMS Learner Information Package Specification (IMS LIP) (IMS LIP, 2008) and eduPerson (eduPerson, 2007) are three of the most important examples of such standards, which are developed for different purposes. They describe information about a user within several categories. In addition, RDF models are used to describe learning resources, such as RDF bindings of Learning Object Metadata (LOM) (Nilsson et al., 2003).

The IEEE PAPI is a data interchange specification developed for communicating between different systems. It describes information about the learners and this is represented in six categories: *Personal information* (information about the learner, i.e. student's name, address, etc.), *relations information* (learners' relationships with other people, i.e. teacherof, classmate), *preference information* (learner's preferences, i.e. language), *performance information* (measured performance of the learner, i.e. grades, certificates), *portfolio information* (previous projects and works), *security information* (public and private keys).

The IMS LIP covers information about a person that is similar to a CV. It is mainly developed for recording lifelong achievements of learners and the transfer of these records between institutions. The IMS LIP consists of eleven categories: *Identification* (biographic and demographic data about the learner, i.e. name, address), *goal* (information about learning, career and other objectives), *Qualifications, Certifications and Licenses* (list of qualifications, certificates and licenses from recognized authorities), *activity* (learning related activities, includes training, work experience, etc.), *transcript* (institutionally-based summary of achievements), *interest* (describes recreational hobbies and activities), *competency* (describes skills, knowledge and abilities), *affiliation* (membership of professional organizations), *accessibility* (describes language capabilities, disabilities, eligibilities and learning preferences), *securitykey* (passwords and security keys assigned to a learner) and *relationships* (relationships between core data elements).

eduPerson is designed to facilitate communication between higher education institutions, particularly for exchanging information about people between US universities. This specification is released jointly by Internet2[17] and EDUCAUSE[18]. In eduPerson, information is organized into object classes and attributes. The specification covers very detailed information about the person and the organization they belong to. The latest version of the eduPerson Object class was released in December 2007 (eduPerson, 2007) and it contains 43 attributes, which are classified in two categories: *General Attributes* and *New Attributes*. General Attributes hold information about a person in higher education (i.e. name, address, security settings, etc.). The second category is New Attributes, which is generated to facilitate collaboration between institutions (i.e. affiliation, entitlement, authentication ID, relationships to the institution, etc.).

In addition to commonly used user model specifications, ontology-based user modeling approaches have been studied by many authors. (Razmerita et al. 2003) presents a generic ontology-based user modeling architecture, which is named Ontologging for knowledge management systems. The user ontology is implemented using Semantic Web technologies and structured on an extended IMS LIP specification. The user

[17] www.internet2.edu [last accessed, 17/6/2008]

[18] www.educause.edu [last accessed, 17/6/2008]

model can be updated explicitly by the users and implicitly by the intelligent services. (Dolog and Nejdl, 2003) discusses a user model that is a combination of the IMS LIP and IEEE PAPI to provide semantically improved personalization services in learning systems using a peer-to-peer environment. This system extends the ontologies used with a *calendar* concept, which holds information about any appointments and events the user has to attend. (Yudelson et al., 2005) proposes a meta-ontology (a top-level classification) for user modeling. This paper provides a comprehensive user ontology for providing AH. In addition, Friend-Of-A-Friend (FOAF) (Dumbill, 2002) can be used for the purpose of adaptation as presented in (Ounnas et al., 2006). In this work, FOAF is extended for building and representing learners' social communities. Semantic Web-based models are also employed to improve AH. (Kravcik and Gasevic, 2006) introduces an enhanced AH application model to improve the interoperability of the components of the AH model using the Semantic Web technologies.

4.7 Discussion

The IMS LIP and IEEE PAPI are well known user modelling standards and have been used by many systems (Razmerita et al., 2003; Dolog and Nejdl, 2003). The eduPerson specification is also widely deployed. Although these standards can be applied to any domain, they are mainly developed for learners and they do not contain data about the user's browsing interests, browsing goals and browsing strategies. IEEE PAPI, IMS LIP and eduPerson are very useful for student-based adaptation in educational hypermedia. However they are not very efficient for Web-based IR adaptation or adaptive presentation of Web content. Because, these specifications require very detailed information about a user and an average Web user is not willing to provide such information. According to the study of (Schiaffino and Amandi, 2004), ordinary Web users favour simple feedback mechanisms which requires less interaction with them for providing explicit feedback to the personalization. For instance, preference-based adaptation or interest-based adaptation requires less interaction with users comparing to complex feedback mechanisms which demands time and effort. In addition, in our opinion, there is a need for generic user profiles, which model the user's interests, goals and browsing strategies, as well as being adaptable to different Web domains.

Achieving open-corpus AH is very challenging because of the dynamic nature of the information space. We believe that Semantic Web technologies can offer solutions to overcome these problems. For instance, information extraction and annotation technologies can be used to infer the context of any Web page and can be dynamically annotated with metadata associated with appropriate ontologies. Ontologies provide a vehicle for structuring the gathered data and the metadata can be related to ontology-based user models at run-time. In addition, Semantic Web technologies provide reasoning capabilities, which can be utilized to perform adaptations on diverse datasets.

4.8 Chapter Summary

This chapter summarized adaptation mechanisms used in AHSs, application areas of AH, AH technologies and methods, and outlined state of the art in AH research. AH research can be divided into three categories: pre-Web AH, Web-based AH and Semantic Web-based AH. Pre-Web AH is studied in closed environments and most applications are in the educational hypermedia domain. With the growth of the Web, Web-based AHSs have emerged. Again most applications are in the educational hypermedia domain. In Web-based AH, open-corpus hypermedia systems, such as Microcosm can be used to support adaptation. The main problems in Web-based AH are interoperability and reusability. Most of the applications use their own standards (languages, rules, etc.) to represent user models and domain models. As a result of this, information cannot be shared or reused at cross applications. Additionally, it is difficult to make changes to the domain model, which requires re-organization of the model. To overcome these shortcomings, Semantic Web-based metadata can be utilized.

The Semantic Web enables interoperable metadata about users, which can be shared and reused by different applications. In this context, different user modelling standards are developed, for instance IEEE PAPI, IMS LIP and eduPerson. However, these standards are basically designed to support information exchange between institutions and not suitable for Web-based personalization. Furthermore, the Semantic Web technologies can offer solutions to achieve open-corpus AH on diverse Web domains. In the next chapter, related work in semantic portals, Semantic Web browsers and semantic annotation will be given.

5 Related Work

In this chapter related work in semantic portals, Semantic Web browsers and semantic annotation research is discussed.

5.1 Semantic Portals

In this section, first we explain the definitions of Web portals and semantic portals. Then, we review research in semantic portals. Also the problems and limitations of existing semantic portal approaches are laid out.

5.1.1 What is a Web Portal?

A Web portal is a platform for information presentation and exchange over the WWW. It provides a point of access into a collection of information about a domain in an organized single site (Jin et al., 2001). According to (Sidoroff and Hyvonen, 2005), portals can be categorized into three main groups based on their functionality: service portals, community portals and information portals. Service portals collect a set of services together and address wide audiences, for example Yahoo[19] provides various such services to their customers. Community portals are designed for community members to support and facilitate the activities of community of interest (Spyns et al., 2002). Information portals contain huge amounts of information about a domain or contain an organized collection of hyperlinks to other resources (Reynold et al., 2004). Information portals can range from broad to specific domains. An example is the fish species portal[20].

[19] http://www.yahoo.com [last accessed, 17/6/2008]

[20] http://www.fishbase.org/search.php [last accessed, 17/6/2008]

Conventional Web technologies are employed in the implementation of a Web portal. These technologies have well-known limitations; information access, search, integration and sharing are difficult and time-consuming tasks (Lara et al., 2004). In this context, the Semantic Web is a possible solution to overcome the limitations of standard Web technologies.

5.1.2 What is a Semantic Portal?

The term "semantic portal" refers to organized web sites that contain collections of semantically structured information. Ontologies are used for structuring, accessing, sharing and the presentation of knowledge. In this sense, Web portals that are implemented using the Semantic Web technologies are known as semantic portals. They can be any type of Web portal, such as service portals, information portals or community portals. The contents of the semantic portals are represented by metadata using ontologies. Since, metadata is machine processable, the contents of semantic portals are not just limited to human consumption but accessible by software agents.

The aim of the semantic portal approach is to solve the integration and information sharing problems of Web portals using machine-processable metadata. Additionally, semantic portals try to improve information access (browsing and searching) using Semantic Web technologies. In the rest of this section, we present a selection of semantic portals.

5.1.3 State-of-the-Art Semantic Portals

This section discusses the state-of-the-art semantic portals. In table 5-1, different features of these portals are summarized.

The SEAL (SEmantic portAL) framework was introduced for providing and accessing information at a portal (Maedche et al., 2001; Maedche et al., 2002). For the case study of this framework, the AIFB Web site is used[21]. The information in the portal is generated by using RDF CRAWLER. The main functions of SEAL are navigational views, semantic search and semantic personalization. The content of the portal can be

21 http://www.aifb.uni-karlsruhe.de/english [last accessed, 17/6/2008]

presented as HTML (for humans) and/or RDF (for agents). Semantic searching is based on comparing the search query with the knowledge base by using semantic inferencing, and then ranking the results according to semantic similarity. Semantic personalization is based on the users' semantic bookmarks and semantic logfiles (tracking of the users' access patterns). A semantic bookmark basically contains predefined query formulas, and users can personalize this bookmark by giving names, choosing stylesheets, or marking it as the starting point. In addition, semantic logfiles are used to track the ontology concepts visited, in order to evaluate and maintain the ontology.

The KAON Portal[22] is a tool that enables the building of ontology-based Web portals, based on the SEAL framework (Ehrig et al., 2002). KAON generates an ontology-based portal by syndicating information from HTML, XML, relational DB and RDF sources, and by means of forms. The main disadvantage is that whenever information is updated from the sources or the ontology is modified, the modifications are not seen at run-time, but have to be regenerated. The main functions of the portal are to provide semantic search and navigational views. Semantic search is based on the SEAL approach. Portal contents can be accessed by users as HTML, and by agents as RDF, which is based on SEAL. This approach is more focused on the creation and management of ontologies. User based personalization is not provided.

OntoWebber[23] is a tool for building data-intensive Web sites. As a demonstration, the Semantic Web Community Portal was built to exchange and share knowledge (Jin et al., 2001). The content of the portal is created by collecting data from heterogeneous Web sources and converting them into RDF. The main feature of OntoWebber is the modeling of the Web site. Domain modeling is used for the construction of the ontology. Site view modeling is a process for the modeling of navigation, content and presentation. In this process, a privileged user (administrator) organizes the links, the contents, and the order of the presentation. These presentation and navigational designs are produced independent of the ontology. In personalization modeling, different models of site views are used to assign users to different groups (i.e. customization). Different presentations are provided based on the groups. In this approach, all personalization features are assigned by the administrator and the administrator is

[22] http://kaon.semanticweb.org/ [last accessed, 22/12/2008]

23 http://semanticweb.org/ [last accessed, 17/6/2008]

responsible for the maintenance of the models and the user groups. Thus, the presentation is completely controlled by the administrator, which is not realistic for a portal, which contains huge amounts of information and numbers of users. Also the users do not have control of their profiles.

The OntoWeb portal[24] is a dissemination tool for the EU-funded thematic network OntoWeb (Spyns et al., 2002). The main functions of the portal are content provision, browsing and querying. In content provision, information can be inserted into the portal in two ways: by means of forms and by syndicating contents which are annotated by common ontology from the external Web resources. The OntoWeb portal supports a syndicator mechanism, which provides a workflow for publishing information (private, pending and public) and it is supervised by a privileged user. In addition, the OntoWeb portal provides two types of querying: term-based and template-based. No adaptation is provided for end-users.

ODESeW is a framework for generating knowledge portals (Corcho et al., 2003). It serves as an intranet and extranet platform for the EU-funded project Esperonto[25]. ODESeW supports the import and export of information in different formats. ODESeW's main features are content editing/provision, presentation, searching, and querying. In content editing/provision, ODESeW allows inserting, updating and removing of class instances and their attributes and their relation instances, based on read and write permissions of the users. However, the interface is difficult for inserting and removing relation instances. In presentations, different visualizations are provided, based on permissions of the intranet and the extranet users. Moreover, keyword-based and ontology-based searching is supplied for the querying. However, related links between information items are not emphasized and in ontology-based search, it is difficult to enter relationship values. An extended version of this framework, called ODESeW 2.0, was also released in the combined work of Esperonto, Knowledge Web and OntoGrid projects (Corcho et al., 2006). In the new version, a User Ontology is used to specify read and write permissions to different parts of the data model and two extensions are added to the architecture: an external information gateway and a notification service. The external gateway is used to feed the data model with

24 http://www.ontoweb.org/ [last accessed, 17/6/2008]

25 http://www.esperonto.net/ [last accessed, 17/6/2008]

information from external resources, and the notification service is used to send asynchronous messages about changes in the data model. The majority of work on ODESeW is concentrated on information sharing between project members.

OntoWeaver is a framework that enables the design and development of customized data-intensive Web sites (Lei et al., 2004; Lei et al., 2004b). Different site views and presentation layouts are defined for the presentation of the contents by using site ontology and presentation ontology. The main feature of OntoWeaver is customization. Users are modeled by a User Ontology and customization rules are used to support customization. Different site views and layouts are assigned to users by the administrator. The presentation is changed depending on the context and the user. In addition, OntoWeaver supports content provision using templates and provides searching using forms. As with OntoWebber, this approach provides a customization framework that allows site designers to provide complex presentation styles and layouts for user groups or individuals. However, the presentation is again completely controlled by a privileged user, and users do not have any control of their profiles.

MuseumFinland[26] is a semantic portal for Finnish Museums (Hyvonen et al., 2004; Hyvonen et al., 2004b). It is an application of the Semantic Web portal generator ONTOVIEWS (Makela et al., 2004). The ontologies and instances for MuseumFinland are created in a semi-automatic way (Hyvonen et al., 2004c). The museums first transform their collections to XML. Information in XML is transformed to XML Schema and then RDF. For these transformations, a semi-automatic tool is used. For manual editing and updating of ontology and instances, the Protégé ontology editor is used. However, the system does not have a distributed maintenance interface. MuseumFinland’s main features are a combined keyword and multi-facet search, and recommendation links. Recommendation links are generated using rules. In addition, the user interface can be adapted to different devices, such as mobiles or PCs. However, user-based personalization is not provided.

The Rewerse portal[27] is a consequence of the work in the Rewerse project (Abel and Henze, 2005; Brunkhorst and Henze, 2005). It is based on SWED-E[28] portal technology

---

26 http://www.museosuomi.fi/ [last accessed, 17/6/2008]

27 http://personal-reader.de:8080/portal/ [last accessed, 17/6/2008]

with the dual aims of reasoning and customization. In the Rewerse portal, the content is updated by scanning known data sources for new or changed metadata, as with SWED-E. There is no distributed Web interface for the maintenance of the contents. Faceted search and personalization are the main features of the portal. Personalization is based on calculating the browsing and professional distances of the on-line users and presents predefined filters. For example, ontological information about Web resources is used to calculate the browsing distance, and the nodes representing authenticated users are used to calculate the professional distance. Then, the distance is visualized with a *radar* applet. However, personalization is not very helpful for improving the browsing facility of the users since it only displays information about related users.

REASE (Repository of Semantic Web Learning Units) has been developed as a part of the KnowledgeWeb and Rewerse projects (Diederich et al., 2007). This repository is intended to support the creation and sharing of knowledge for higher education in the areas of the Semantic Web and ontology technologies. The contents of the repository can be updated by the users by adding new materials (tutorials, lectures, etc.). The main features of the repository are ontology-based search, browsing, and collaborative personalization. The ontology hierarchy is used to support the browsing. In the ontology-based search, the system provides valid values for relationship attributes. Search results also can be ordered based on the collaborative ranking, alphabetical order, creation time of the document, etc. The system also supports collaborative personalization. The users of the repository first need to register with the portal. After registration, users can save material they are interested in to their profiles. The users can also update their profiles from within the interface. The personalization is done by allowing users to order the search results or the browsing material using the collaborative ranking.

28 http://www.swed.org.uk/ [last accessed, 17/6/2008]

Table 5-1: Comparison of different features of semantic portals

	SEAL	KAON	Onto Webber	OntoWeb	ODESeW	OntoWeaver	MuseumFinland	Rewerse	REASE	SEMPort
Content editing/ provision	Using RDF crawler and OntoEdit ontology engineering workbench	Syndicating from web pages, DB, RDF files and using Web forms. Do not operate in real-time	Collecting data from Web pages using wrappers and converts them into RDF	Syndicating mechanisms and the use of Web forms. Do not operate in real-time	Allows import/export of data. An interface to edit data and an external info gateway	Through knowledge acquisition forms (templates)	Semi-automatic tool to convert XML data to RDF, Protégé (edit/update)	Based on SWED-E wrappers (scanning known data sources)	Updated by users by using Web forms	Through RDF file aggregator web interface, Protégé (edit/update) and editing/provision Web interface (editing). Operate in real-time
Search	Ontology-based. Ranking based on similarity to query	Ontology-based	✘	Term-based and template-based	keyword based and ontology based	Search forms	Combined keyword and multi-faceted search	Faceted search	Ontology-based	Ontology-based using transitive and rule-based reasoners
Links to related pages	✘	✘	✘	✘	✘	✘	Recommendation links to related pages using rules	Links to related pages using bookmark ontology	✘	Links to relevant resources using reasoning and rules
Inference	F-Logic based during search and navigation	KAON datalog engine, OWL reasoner (KAON DL)	An inference engine is attached to navigation module	Based on SEAL (F-Logic)	Prolog inference engine	Jess inference engine to provide customization using rules	SWI-Prolog inference engine for navigation and search	Rule-based reasoner	Not specified	Jena owl and rule-based reasoners for navigation and search
Personalization	Semantic bookmarks (predefined queries)	✘	**Customization** based on different user types and rules	✘	**Customization** based on a user model	**Customization** based on user ontology and customization rules	**Device-based** adaptation (i.e. to PCs, mobiles)	Displays distances of online users and predefined filters	Collaborative ranking based ordering during navigation and search	A user model for supporting AH (link sorting, link annotation, adaptive presentation)

5.1.4 Discussion of Semantic Portals

The process of content provision in semantic portals is a difficult task, especially where the content is continuously changing. The problems associated with content provision are twofold: the newly-added content cannot be seen without restarting the server, and Web interfaces are difficult for the insertion of information. As can be seen from the Table 5-1, many semantic portals use syndicating mechanisms (Web crawlers) and Web forms to collect and publish data. Therefore, there should be easy-to-use, reusable content editing and provision mechanisms, which will help content providers update the information in real-time.

On the other hand, semantic portals provide access mechanisms to the information, such as search and navigation. In most of these cases, navigation is not very effective; the same page content and the same set of hyperlinks are shown to all users, and hyperlinks to similar pages are often not presented. However, different users have different browsing needs and the content should be adapted according to those needs. Most of the semantic portals do not provide, or provide very limited, adaptation to the end-user. For instance, many semantic portals provide customization (SEAL, Ontowebber, OntoWeb, OntoWeaver, Rewerse) as shown in Table 5-1, but only two of the semantic portals provide personalization, such as MuseumFinland (device-based personalization) and REASE (collaborative filtering based on ordering information items). On the other hand many semantic portals do not present links to relevant pages as presented in Table 5-1. Only MuseumFinland and Rewerse provide hyperlinks to related Web resources. Generally, portals contain huge amounts of information and users have difficulty in finding relevant information or often have the feeling of being lost. Thus, the contents of the portals should be adapted to the particular needs of the user and the linking between information items should be enriched to help to improve the navigation by the user.

5.2 Semantic Web Browsers

In this section, first we define what we mean by a Semantic Web browser. Then, related research in the field of Semantic Web browsers is discussed.

5.2.1 What is a Semantic Web Browser?

Semantic Web browsers are designed to provide an interface to RDF. RDF is designed for machines to process information and does not have a visual presentation model for humans. The users and developers of the Semantic Web want to explore and analyze RDF data in a human-friendly form. Hence, tools and applications are developed to allow exploration and traversing of RDF resources with user-friendly interfaces (Berners-Lee et al., 2006b). In this thesis we call them *Semantic Web data browsers*. Semantic Web data browsers can be Web-based or desktop-based and supply presentation mechanisms for RDF. In these systems various presentation paradigms are adopted (i.e. node-link diagrams, table-like layouts, box-like layouts) and special user interfaces are developed (i.e. tree-like structure, faceted browsing).

5.2.2 Related Research in Semantic Web Browsers

In the developing field of the Semantic Web, it is impossible to complete a comprehensive survey of Semantic Web browsers. However, in this section, we briefly summarise the related work and known tools and applications. We divided Semantic Web browser research into two categories: Web-based Semantic Web browsers and desktop-based Semantic Web browsers. Web-based Semantic Web browsers are emerged from different visions and purposes, thus we further divided this category into three groups: Semantic Web browsers for visualizing RDF, Semantic Web browsers with special user interfaces and Semantic Web browsers for supporting end-users. In the following sub-sections, we outline the research in all fields.

5.2.2.1 Web-based Semantic Web Browsers

Semantic Web Browsers for Visualizing RDF: The main design goal of these Semantic Web browsers is presenting RDF data and allowing the traversal of a web of RDF resources in a user-friendly manner (visualize/explore RDF data). Different RDF browsers use different presentation designs (e.g. table-like layouts or box layouts) for presenting RDF content. In short, these browsers enable exploration and analysis of RDF in a user-friendly manner. Examples of Web-based RDF browsers are Nodester (Rutledge et al., 2005), Longwell[29] and SemantExplorer (Scerri et al., 2005). Nodester

[29] http://simile.mit.edu/wiki/Longwell [last accessed, 22/12/2008]

is a generic RDF browser that allows navigation of any RDF repository. Longwell is a Web-based faceted RDF browser implemented within the SIMILE project[30]. It allows visualization of RDF with user customizable facets. In Longwell, the RDF comes from back-end triple stores, which are called *semantic-banks*. SemantExplorer combines graph viewing with attribute-value viewing (item descriptors) for helping both beginners of the Semantic Web and Semantic Web developers to explore RDF.

The Semantic Web browsers Haystack (Quan and Karger, 2004) and Piggy Bank (Huynh et al., 2005) are also examples of Web-based RDF browsers; they collect and create RDF from Web documents and allow users to navigate the metadata. Haystack and PiggyBank were developed within the SIMILE project. The Haystack RDF browser aims to integrate RDF from multiple arbitrary locations and present them in a human-friendly manner. A haystack is a collection of metadata and a person's haystack is a collection of metadata that is collected from all the information the person has come across. In this way, metadata is stored and the unified data is presented to the user. It provides three different views to the users: the browse view (user-friendly); the debug view (allows editing RDF); and the all information view (presents all details of the RDF). Haystack also supports other functionalities, for instance the presentation of the collection of RDF resources, the bookmarking of these collections, the storage of collections and the customization of this data. Recently, the Longwell presentation model was integrated into Haystack. Piggybank is an extended version of Haystack. In PiggyBank, metadata can be created by users while they browse the Web and is stored in a database called a Semantic Bank, where metadata can be shared by users.

The RDF browsers which use resolvable URIs to provide point-and-click access to metadata are known as *linked data browsers*. Linked data browsers also fit into Web-based RDF browsers, since they provide visualization and exploration of decentralized RDF resources. A linked data browser usually operates as follows: it takes a URI as input, looks up the URI over HTTP protocol (dereference) and shows the metadata in a user-friendly form (such as tables, box-like layouts, etc.). Users can explore more data by clicking particular resources and new resources will be dereferenced at the user's request and presented in the same way. Tabulator (Berners-Lee et al., 2006b), Disco

---

[30] http://simile.mit.edu/ [last accessed, 15/9/2008]

(Bizer and Gauss, 2007), OpenLink[31], BrownSauce[32], Marbles[33], Zitgist[34], ObjectViewer[35] and Humboldt (Kobilarov and Dickinson, 2008) are some examples of linked data browsers. One of the common properties of all of linked data browsers is that they are generic in terms of supplying presentation for diverse linked data. On the other hand, some linked data browsers have bookmarklets[36], particularly Tabulator and Disco. A bookmarklet is a bookmark bar added to the browser so that whenever a webpage contains a link header element that refers to "application/rdf+xml", the user can use the RDF browser to display RDF content. The problem of linked data browsers is that if a Web resource does not have machine processable metadata associated to it, the user cannot explore the metadata of the Web page. Also, with linked data browsers, users can browse metadata and Web documents separately. In our opinion, metadata could be more useful if it is presented within the page contents.

Tabulator is an AJAX-based Web application which works with the Mozilla Firefox Web browser. It adopts a nested box-layout. The Disco RDF browser is designed as a server-side application and displays information in a table-like layout. Tabulator and Disco have bookmarklets which enable users to launch those browsers from their standard Web browsers when a URI is available at the Web page. OpenLink is another AJAX-based RDF browser designed by Virtuoso. OpenLink also uses table-like presentation, whilst the BrownSauce RDF browser adopts a faceted browsing interface using a table-like layout. Marbles is a server-side application that uses Frensel lenses and formats (Pietriga et al., 2006) which is a vocabulary for formatting and presenting RDF for displaying content. Different to other RDF browsers, the Zitgist RDF browser can display data coming from diverse data sources, such as from dereferenceable URIs, zitgist's internal database or on the fly conversation of data (i.e. microformats, RDFa, HTML metadata, etc.). The Zitgist presentation layout is table-like format. ObjectViewer provides simple graphs for the visualization of linked data. The Humboldt linked data browser aims to combine different presentation paradigms in one browser; it combines faceted browsing and single object presentation in one screen.

---

[31] http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html [last accessed, 22/12/2008]

[32] http://brownsauce.sourceforge.net/ [last accessed, 22/12/2008]

[33] http://wiki.dbpedia.org/Marbles?v=ypu [last accessed, 22/12/2008]

[34] http://dataviewer.zitgist.com/ [last accessed, 22/12/2008]

[35] http://objectviewer.semwebcentral.org/ [last accessed, 22/12/2008]

[36] http://www.w3.org/2006/11/tabulate.html [last accessed: 26/6/2008]

Semantic Web Browsers with Special User Interfaces: These approaches extend the presentation paradigm of RDF with specialized user interfaces widgets (e.g. calendar data, tree structures or interaction capabilities). For example, Tabulator incorporates calendar views, map views, timeline views and SRARQL query interface.

On the other hand, mSpace (Schraefel et al., 2005), the CS AKtive project (Schraefel et al., 2004) and Flink (Mika, 2005) support domain-specific user interfaces for a particular RDF database (known as triple store). For instance, one application of mSpace is a classical music database for classic music pieces and composers. Users can view same information in multi-dimensions; each column represents one dimension, which contains different properties from the repository. Users can sort or swap columns and information can be presented in multi-views. The CS Aktive project is a Web-based application which presents information in the field of computer science (i.e. people, projects and research) from a triple store. In this approach, information is presented in different panes according to the selected area-radius from the map. RKBExplorer[37] is newer generation of CS Aktive, which is developed within the ReSIST project (Glaser et al., 2008). It combines information from heterogeneous sources (resolvable URIs, personal Web pages, databases) and stores this data to a triplestore for presenting metadata in a unified multidimensional space. The RKBExplorer also supports co-referencing. The Flink project adopted a different presentation paradigm for allowing browsing and analyzing of a database of people. It presents maps of interconnected individuals and users can navigate them in a Web-like structure. The systems described above generally provide application-specific presentations, hence the interfaces are richer and more powerful than generic RDF visualization tools.

Semantic Web Browsers for Supporting End-Users: The final group of Semantic Web browsers are designed mainly to support users in the browsing and interpretation of Web pages using Semantic Web metadata. Example systems are COHSE (Carr et al., 2001) and Magpie (Dzbor et al., 2003; Domingue et al., 2004; Dzbor et al., 2007).

COHSE is a project that tries to define and deploy conceptual open hypermedia link service. COHSE combines the distributed link service architecture with a conceptual

[37] www.rkbexplorer.com/ [last accessed, 2/3/2009]

model to provide conceptual open hypermedia. There are several implementations of it, such as an Internet Explorer Web browser add-on, a proxy-based rewriting server and a server-based portal implementation. COHSE reconstructs the page visited by the user by using the ontological relationships and distributed link services; then an ontology-driven lexicon is used to add links to arbitrary Web pages. Predefined ontologies are used to form a thesaurus for the pages. The links are separated from the Web pages and stored in conceptual link bases. Then link services are employed to use relationships defined by the ontologies to generate hyperlinks to related pages. The demonstrations of the COHSE are undertaken on the Java Sun systems Web portal. Recently, personalization and customization ideas have been proposed by COHSE developers but these ideas have not been implemented (Yesilada et al., 2008).

Magpie is a Semantic Web browser that provides mechanisms for browsing and making sense of information on the Semantic Web. Magpie acts as a complementary knowledge source, which a user can call upon to gain access to background knowledge relevant to the Web resource. This is achieved by associating an ontology-based semantic layer to the Web resources; the semantic layer automatically associates meaning to the pieces of information found on a Web page using ontologies. An appropriate ontology can be chosen by the user from a list of ontologies that are known to the tool. Magpie also allows tracking of the user's browsing history using semantic browsing log files. Semantic browsing log files can trigger other services, which are called collector services. Collector services collect items from the user's browsing session using ontology-based filters and the concepts that are visited by the user during a browsing session are recognized and grouped together at the right pane of the browser (i.e. people, projects, organizations). Collectors can also provide links to related knowledge using semantic bookmarks. Semantic bookmarks estimate queries using ontology-based filters and present links to relevant knowledge. It should also be noted that the semantic browsing log files are stored to a semantic logfile knowledge base. In addition to these, users can highlight concepts of interests in a page using the Magpie toolbar and the underlying ontology, as well as asking for related services through the Magpie interface.

The more recent version of Magpie is PowerMagpie and it uses a different approach (Gridinoc et al., 2008). In PowerMagpie, users are not required to select a pre-defined ontology. The occurrences of linked data instances and classes are dynamically found by using the Watson search engine[38]. Users can explore these data from the sidebar. PowerMagpie brings semantic interpretation to Web browsing; users make sense of available semantic metadata using PowerMagpie. However, in this thesis our research aim is different; our aim is to guide users to relevant resources using linked data and goal services, in addition adapt the information using personalization.

COHSE and Magpie provide different services to users based on the metadata of the page, whether such metadata exists or not. Hence they provide Information Extraction (IE) and semantic annotation to extract information from Web pages. The difference between these browsers and the Web-based RDF browsers described previously is that they are more user-oriented. The aim is to provide rich services (i.e. inferred links, links to related resources, etc.) to the user using metadata.

5.2.2.2 Desktop-Based Semantic Web Browsers

Desktop-based Semantic Web browsers allow the visualization of RDF with different presentation formats, such as directed labelled graph or tables. They also aim to display RDF in a user-friendly manner as do Web-based RDF browsers aim. IsaViz (Pietriga, 2006), Welkin[39] and RDFAuthor (Steer, 2003) are examples of desktop-based RDF browsers and they represent RDF as node-link graphs. However, this kind of presentation is difficult to handle with large RDF graphs.

5.2.3 Discussion of Semantic Web Browsers

In this thesis, our main intention is to support browsing using Semantic Web technologies and AH. Therefore our definition of a Semantic Web browser should provide benefits to users using Semantic Web content whether such content is available or not. Additionally, we believe that the adaptation of hyperlinks to the current task and/or interests of the user may improve browsing and is needed to accomplish personalization in open Web environment. This can be achieved through the use of Semantic Web content. Links to relevant resources can be provided and all information can be personalized according to the user's current needs (user profile). In addition, in

[38] http://watson.kmi.open.ac.uk/WatsonWUI/ [last accessed, 2/2/2009]

[39] http://simile.mit.edu/welkin [last accessed, 22/12/2008]

our system users are not required to wholesale adopt the vision of the Semantic Web, since they can use standard Web browsers with the seamless added support of Semantic Web metadata.

In our opinion COHSE and Magpie are close systems to our approach. They use semantic metadata to provide related information to the users. However, Magpie paid little attention to the user's role and do not supply adaptive links or content. In addition, both systems use static databases for creating semantic content, storing information and generating hyperlinks. In our work, we reuse linked data to create semantic links.

5.3 Semantic Annotation

Before discussing related work in semantic annotation research we first want to clarify the meanings of IE, annotation and semantic annotation.

5.3.1 What is Information Extraction (IE)?

IE is a type of information retrieval which is used to extract structured information (i.e. classes of events or relationships) from unstructured documents (Grishman, 1997). IE is needed because enormous amounts of information exist in unstructured documents. It is very difficult and time consuming to analyze and manipulate it. If data can be represented in a structured form, then it can be automatically analyzed and used. To address the IE problem, a variety of systems and techniques have been developed, such as statistical methods, hidden Markov models, probabilistic context-free grammars and rule-based methods that utilize some form of machine learning.

5.3.2 What is Annotation?

> **Annotation** [noun]: A note by way of explanation or comment added to a text or diagram.

In the computer context, annotation means attaching of a set of instantiations to an HTML document. (Bechhofer et al., 2002) categorized three types of annotation: textual annotation, link annotation and semantic annotation.

Textual annotation is the process of inserting notes and comments to resources. This kind of annotation is supported by Annotea (Kahan et al., 2001; Koivunen, 2005), whereas richer annotation types, such as commentaries can be marked. Link annotation extends the notion of text annotation, whereas it allows the addition of links to arbitrary documents. One such examples of is the Distributed Link Service (Carr et al., 1995). Finally, semantic annotation is the inclusion of rich semantic information to the page content based on ontologies. The main focus of this thesis is semantic annotation.

Web page content is not understandable by machines unless its meaning is expressed in a formal way. Semantic annotation thus provides machine processable meanings about a Web resource. According to (Zhihong and Mingtian, 2003), this is done by committing a Web resource to particular ontologies. (Ding, 2005) also describes semantic annotation as *"…labelling Web page content explicitly, formally and unambiguously using ontologies*". In addition, according to (Kiryakov et al., 2003), semantic annotation is "*a specific metadata generation and usage schema targeted to enable new information access methods and extend existing one*". In short, semantic annotation is used to add metadata to a Web resource using ontologies.

Semantic annotation methods can be grouped into two categories (Scerri et al., 2005): internal annotation and external annotation. Internal annotation embeds the semantic markup inside the HTML document and external annotation stores the metadata in a separate file. Ontobroker (Decker et al., 1999) and SHOE (Heflin et al., 1999) are examples of internal annotations, where the markup is embedded inside the HTML documents, and some processing is required to extract metadata from pages for consumption by semantic agents. On the other hand, typically annotations are distributed in separate files, such as linked datasets. The W3C[40] suggests the use of external markups stored to a file. Recently, the W3C has introduced a new draft for embedding machine-processable data into HTML using RDFa (W3C, 2008d). The idea is to augment human readable data with machine-processable data. In this way, Web browsers can augment the content with machine-processable information.

[40] http://www.w3.org/RDF/FAQ [last accessed, 12/6/2008]

5.3.3 *Semantic Annotation Research*

This section attempts to summarize the main techniques in the field of semantic annotation. A comprehensive survey of semantic annotation tools can be found in (Reeve and Hun, 2005) and (Uren et al., 2005). These surveys categorize semantic annotation tools differently. Reeve and Hun classify platforms based on the annotation method used, such as pattern-based, machine learning-based and multi-strategy based. On the other hand, Uren et al. separate semantic annotation techniques into semantic frameworks and tools. However, both surveys agree on the main different approaches to semantic annotation, such as manual, semi-automatic and automatic annotation.

In manual annotation, a user annotates a document manually using ontologies. For instance, the Amaya browser-editor[41] supports the manual annotation of pages. The drawback is that manual annotation requires trained staff, time and effort and it is prone to errors. In semi-automatic annotation, the text is analyzed to find occurrences of instances and then the recognized instances are related to the corresponding ontological entities. In these systems, human intervention is required to clarify unambiguous terms, so they are not completely automatic. SemTag (Dill et al., 2003) is an example of semi-automatic annotation. In automatic annotation, the systems perform the annotation without the intervention of humans. An example is C-PANKOW (Cimiano et al., 2004).

5.3.3.1 Semantic Annotation Frameworks

Uren et al. discusses two annotation frameworks: the W3C's Annotea project (Kahan et al., 2001; Koivunen, 2005) and CREAM (Handschuh et al., 2001). Annotea is tool for collaboratively annotating Web documents with shared annotations. These annotations can be comments, notes, explanations and other external remarks. It uses an RDF annotation schema for describing annotations. The generated metadata can be stored to a local file or to annotation servers. An example client implementation of the Annotea is the W3C's Amaya browser-editor for annotating documents with RDF. CREAM (Creating RElational, Annotation-based Metadata) is a framework for generating relational metadata comprising class instances and relationship instances (Handschuh et al., 2001). The CREAM annotation framework is based on the following components:

[41] http://www.w3.org/Amaya/ [last accessed, 20/2/2009]

document viewer to visualize the Web page, *ontology guidance* for helping users using an ontology, *crawler* for searching the existing annotations for a semantic instance, *annotation inference server* for querying instances and checking the consistency of metadata, *document management* for managing annotated documents and *information extraction* to semi-automatically annotate Web documents.

5.3.3.2 Semantic Annotation Tools

Manual annotation tools: The Amaya browser-editor is an implementation of Annotea for annotating documents with RDF. Ont-O-Mat-Annotizer is an interactive Web page annotation tool for annotating Web pages with OWL instances, attributes and relationships (Handschuh et al., 2001). Ont-O-Mat-Annotizer is based on the CREAM framework.

Semi-automatic annotation tools: S-CREAM (Handschuh et al., 2002) and SemTag (Dill et al., 2003) are example automatic annotation tools. S-CREAM provides semi-automatic annotation using Amilcare information extraction component and Ont-O-Mat-Annotizer. The Amilcare is a system that learns information extraction rules from manually marked-up rules. SemTag is a platform for large-scale text analytics to perform semantic tagging of large corpora. It has been applied to a collection of approximately 264 million Web pages and generated approximately 434 million automatically disambiguated semantic tags. SemTag annotates Web resources using TAP knowledge base, which contains broad range of lexical and taxonomical information about popular objects, such as music, movies, authors, sports, etc. For resolving disambiguated semantic tags, it utilizes Taxonomy Based Disambiguation (TBD) algorithm.

Automatic annotation tools: C-PANKOW (Pattern-based ANnotation through Knowledge On the Web) is an unsupervised pattern-based approach to categorize instances according to a given ontology (Cimiano et al., 2004). In this approach, a Web page is scanned for phrases that might be instances of an ontology and series of linguistic patterns are applied. Then, google search API is used to find the meaning of an instance according to the number of occurrences of it in search results. C-PANKOW is also integrated to the CREAM framework.

On demand annotation tools: Magpie supports on-demand annotation of un-annotated Web pages from a browser (Dzbor et al., 2003; Domingue et al., 2004). It uses a simple parser that annotates a Web page according to a particular ontology using ontology-driven lexicon. Users can use the annotated Web page for highlighting occurrences of ontology instances.

Other semantic annotation techniques and tools: KIM is a semantic annotation platform for automatic semantic annotation, indexing and retrieval (Kiryakov et al., 2004). It is based on GATE for IE, which recognizes references to entities from the text and matches those references to URIs from a Knowledge Base (KB). Alternatively, new URIs and entity descriptions can be generated. KIM uses an upper-ontology for metadata creation and all metadata is stored to a KB. KIM also offers a server, Web user interface and Internet Explorer plug-in. By using the plug-in, automatic hyper-linking is enabled based on the created metadata of the document. COHSE also annotates Web documents using pre-defined ontologies, annotation wrappers, GATE (i.e. class instances are identified using GATE) or manual annotation[42] (Yesilada et al., 2008). Piggybank utilizes screen scrappers to extract metadata from particular Web pages and stores this data locally together with the tags of the users for later searches (Huynh et al., 2005). Another related project in metadata generation is the OpenCalais[43] project. OpenCalais is a web-service provided by Thomson Reuters for automatic semantic metadata generation. It supplies a programatically accessable API for analyzing text and extracting semantic information from it in the form of entities and relationship instances. Since January 2009, it supports annotation of documents with linked data URIs as well. The extracted data can be returned in variety of formats, such as RDF, microformats (rel-tag, hCard and hCalendar) or JSON. Microformats[44], is a semantic markup technique that integrates data elements into HTML/XHTML/XML files. An example data format for microformats is hcard (for people and organizations) and hCalendar (for calendars and events). In microformats markup is inserted into the HTML document but they do not use a formal ontology.

---

[42] http://research.sun.com/sunlabsday/docs/Talks/tBigideas/4.02_green_horan.pdf [last accessed, 2/3/2009]

[43] http://www.opencalais.com/ [last accessed, 20/2/2009]

[44] http://microformats.org/about/ [last accessed, 2/3/2009]

5.3.4 Discussion of Semantic Annotation Tools

From the survey of semantic annotations, we have seen that there are two forms of annotations: internal and external. Recently, with the introduction of microformats and RDFa, there is a movement and discussion of embedding markup into documents. On the other hand, the new trend of the Semantic Web is linked data, where in this approach annotations are stored separate than documents. To the best of the researcher's knowledge, it is still not clear how embedded markup of the microformats and RDFa will advance the expansion of the Semantic Web metadata.

On the other hand, we found that automatic semantic annotation is very challenging and prone to errors because of the need for co-referencing and resolving ambiguous words. In addition, the survey also showed that many semantic annotation tools and applications are specifically designed for a domain since prior knowledge of the information space is essential. Furthermore, most of the annotation tools use their own vocabulary and pre-created class instances for annotating Web pages with markup. However, this approach is limited by the metadata available to the system. Alternatively, open standard linked data can be used to annotate Web documents with a wide variety of available metadata on the Semantic Web.

5.4 Chapter Summary

In this chapter, related work in the field of semantic portals, Semantic Web browsers and semantic annotation research were discussed in three sub-sections. First, definitions of Web portals and semantic portals were given, then the related work in the field of semantic portals were summarised. From this literature review, we identified problems of the current semantic portals, such as content provision is a difficult and challenging task, links to related information is often not presented and the same set of links and content is shown to all users. To overcome these problems, we propose a semantic portal, *SEMPort*, which will be discussed in chapter 6. Then, we explain the different definitions of Semantic Web browsers which arise from different objectives and discuss the related research in Semantic Web browsers. We divided the research into two groups: Web-based Semantic Web browsers and desktop-based Semantic Web browsers. To enhance existing Semantic Web browsers, we propose a Semantic Web browser, *SemWeB*, which will be discussed in chapter 8. Finally, the definitions of IE, annotation and semantic annotation were given and related work in the semantic annotation research was discussed.

6 SEMPort – A Personalized Semantic Portal

To alleviate the problems of the semantic portals discussed in the previous chapter, we developed a personalized SEMantic Portal (SEMPort), which is called SEMPort (Şah and Hall, 2007; Şah et al., 2007). In this chapter, the design and implementation of SEMPort is described in detail.

For the evaluation of SEMPort, we have used the ECS Course Modules Web Page (ECS CMWP, 2006). However, SEMPort can be easily adapted to different ontology domains with small modifications, since it is implemented independent of ontologies.

In the remainder of this section, we explain our approach using the ECS CMWP for illustration. First, use cases of SEMPort and the system design of SEMPort are explained. Then the domain ontologies, semantic annotation process on the evaluation domain and the user ontology (for representing user profiles within the ECS CMWP) are discussed. The functionalities of the proposed approach are laid out with examples from the ECS CMWP domain. In addition, the reader is referred to the Appendix C for comprehensive walkthroughs of SEMPort.

6.1 Use Cases

Our research intends to create an ontology neutral semantic portal and aims to aid information discovery and navigation in a portal using dynamic linking and personalization. In addition, we purpose to provide easy-to-use content maintenance mechanisms which can operate in real-time. To demonstrate our approach, in this section, we briefly discuss use cases of SEMPort using the ECS CMWP. ECS CMWP is a website that provides information about modules within the School of Electronics and Computer Science. In SEMPort, we used exactly the same content but this data is represented by ontologies. Our aim is to improve information access within ECS CMWP using SEMPort.

In SEMPort, the contents of the portal are presented using ontologies. For supplying navigation, concepts from the ontology are presented at the left-pane of the portal (see Figure 6-1, left pane). When a user clicks onto a concept from this ontology hierarchy, an ontology-based search form and instances of the selected class are shown at the right-pane, which is known as *general view* (Figure 6-1, right pane). If the user is logged into the portal, SEMPort also adapts information to the user. For example, according to the interests of the user, information resources are ordered during presentation at the general view as presented at the right pane of Figure 6-1. Using the general view, users can perform ontology-based searches. Whenever, a user clicks on an instance from the general view, more information is shown in the detailed view (see Figure 6-2). In the detailed view, dynamically generated recommendation links to related instances, inverse links and inferred links are presented in addition to the links coming from ontologies. Furthermore, recommendation links are annotated with different visual cues depending on their relevancy to the user's profile (Figure 6-2).

Using SEMPort, users can navigate the information space using ontology hierarchy, ontological relationships and dynamically generated hyperlinks. Besides, this information is personalized according to user profiles (using background knowledge and interests). We also provide a personalized homepage for supplying personalized access to the interested parts of the portal and a profile editor for updating user profiles explicitly. Furthermore, authorized users can update/edit the portal contents from a distributed content editing/provision Web interface. Updated information can be seen without restarting the Web server in real-time.

SEMPort is also implemented with reusable components and can be adapted to other ontology domains with a low cost. More information about all of the functionalities of SEMPort will be discussed in the upcoming sections of the thesis. In addition, more detailed walkthroughs of the use of SEMPort are presented in the Appendix C.

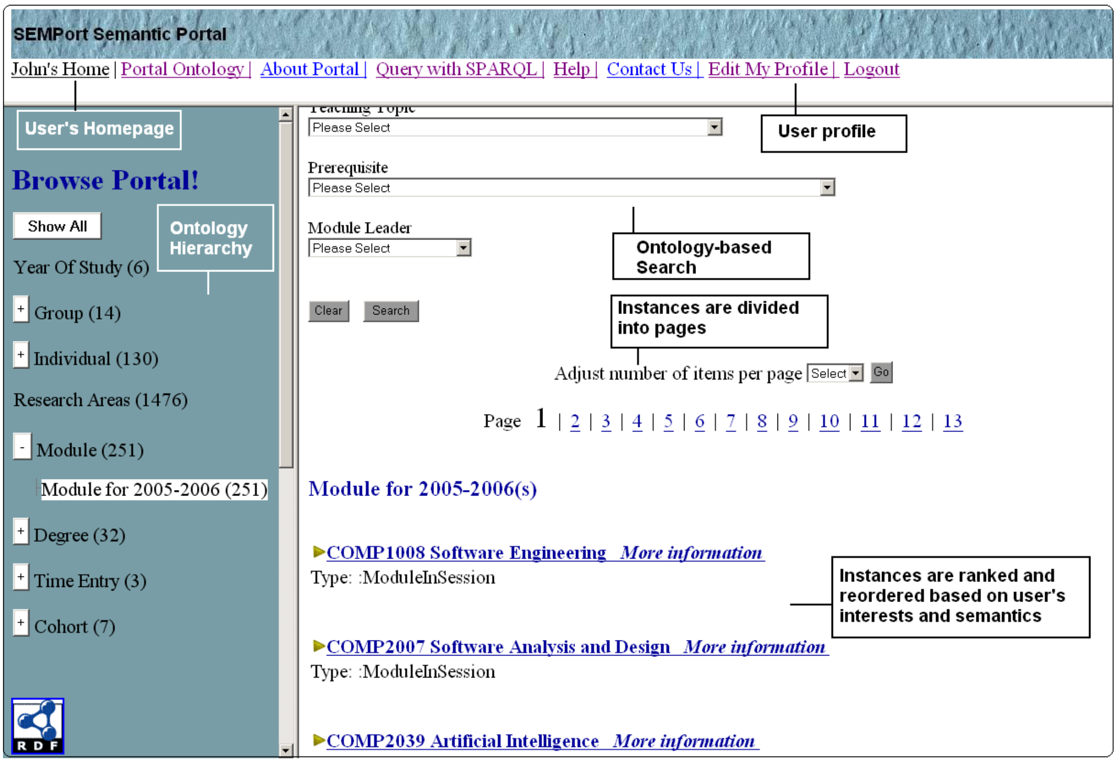


Figure 6-1 A general view from the semantic navigation

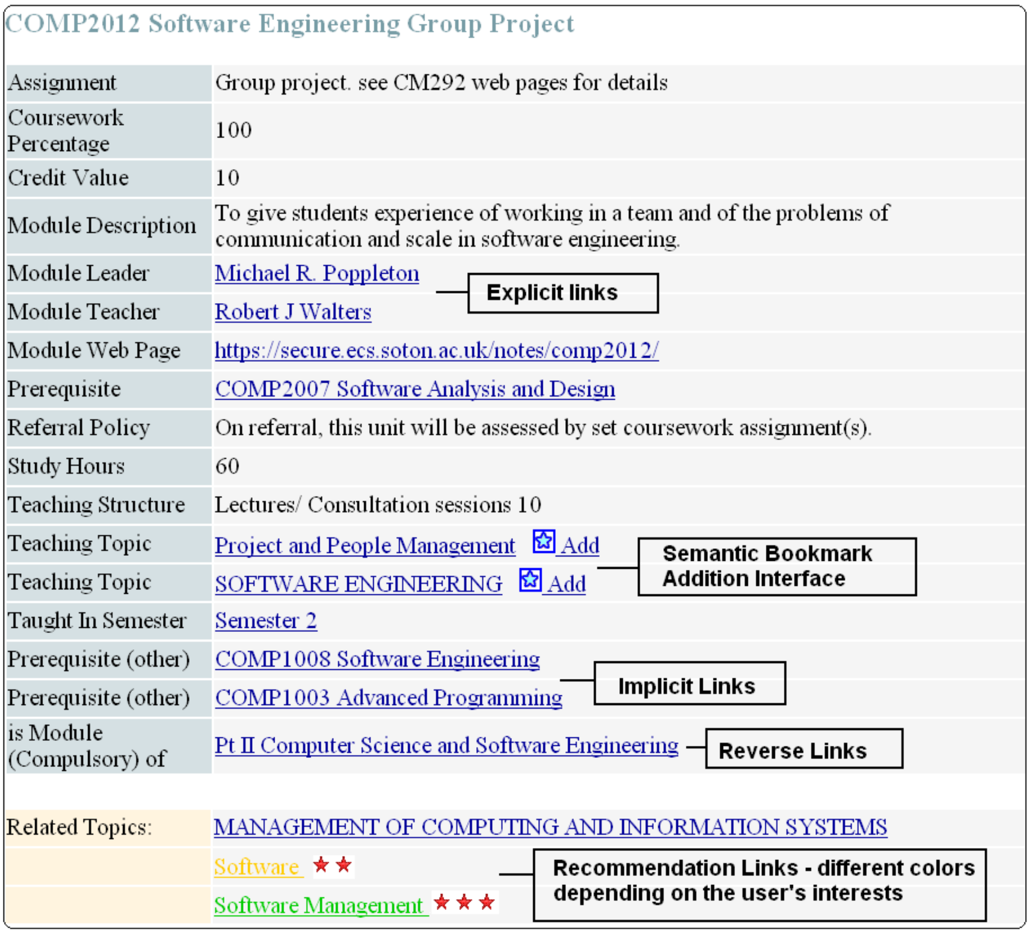


Figure 6-2 A detailed view that illustrates explicit links, inverse links, implicit links, recommendation links and semantic bookmark addition interface.

6.2 System Design

The architecture of SEMPort is shown in Figure 6-3. The ontology and instances are stored in a database, which is known as the Knowledge Base (KB). The KB can be uploaded either by using the Protégé Ontology Editor or by using a Web front-end, and can be maintained by authorized users from a distributed content editing and provision Web interface. Users can access the portal's contents from any ordinary Web browser. Then, they will be provided with semantic navigation according to the ontology domain. For instance, the ontology hierarchy is presented and used to supply navigation for the portal content. When users click onto ontological concepts from this hierarchy, instances of the concept are shown to the users together with ontology-based search. This view is a general view and only instance titles are shown to the user. In this view, users can perform ontology-based searches or can click on to semantic instances. When users click on to the instances of interest to them, then more detailed information is shown together with ontology-based and inferred links (i.e explicit, inverse, implicit and recommendation links). If the user is logged in to the portal, we also personalize links according to the user model by annotating them with visual cues or re-ordering them. In addition, personalized homepages can be shown on the user's request.

In Figure 6-3, the interactions of each SEMPort module is shown. The adaptation module is responsible from the creating user's metadata and maintaining it. An ontology-based search engine queries the KB according to a search query. The inference module infers implicit relations between instances (reasoning about the explicit knowledge). As can be seen, the navigation module is a middle layer between the user interfaces and all other modules of SEMPort. The navigation module amalgamates all information collected from the other modules and presents this information to the user.

With walkthroughs, the interactions of SEMPort modules can be explained in detail as follows: When the user clicks on a concept from the ontology hierarchy, URI of the requested ontology class is sent to the navigation module. The navigation module first invokes ontology-based search module and asks for ontological properties of this class (i.e ObjectProperty, DatatypeProperty). Ontolgy-based search module queries the KB with SPARQL queries and sends properties and available instance values to these properties to the navigation module. Navigation module receives this data and creates an ontology-based search form at the top of the page. Then, the navigation module queries the KB for the instances and sub-instances of the requested ontology class using SPARQL and receives instances from the KB. Then, present these semantic instances using a general view. If the user is logged in, then the navigation module queries the user profile and reorders instances according to the user. When the user clicks on an instance from general view, then URI of the semantic instance is passed to the navigation module. Navigation module first creates a Jena inference model and attaches rules from the inference module, then queries the KB for the instance's properties (ObjectProperty and DatatypeProperty) using SPARQL and receives instance's properties. Finally, all the information and links are presented as a detailed view. Whenever a search is made, then ontology-based search module queries the KB by dynamically regenerating a SPARQL query and finds matching semantic instances from the KB. The results are sent back to the navigation module and it is presented as a general view.

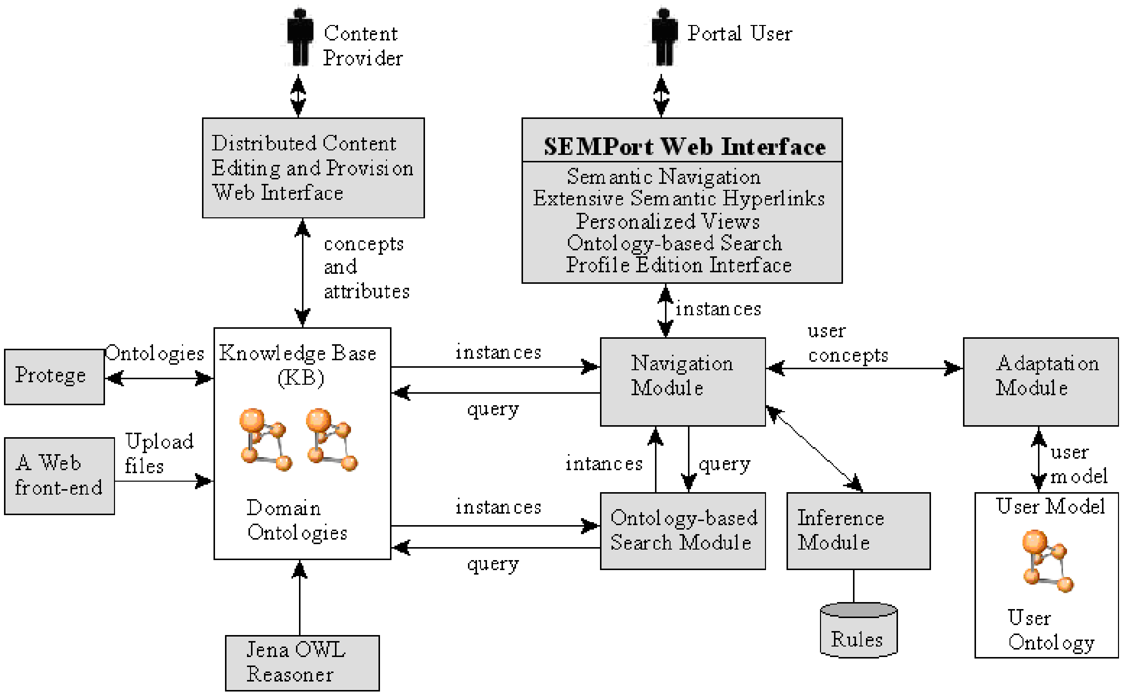


Figure 6-3 The architecture of SEMPort

The Jena Ontology API (Jena, 2008), the Jena OWL reasoner, the Jena rule-based reasoner (Jena Inference, 2008) and a set of rules are the main components of the SEMPort architecture. The OWL reasoner is bound to the KB, which checks the consistency and validity of the knowledge. In addition, the rule-based reasoner is utilized by the inference module for inferring implicit knowledge based on rules. The

Java code for supplying inferencing to SEMPort, and an example of rules are given in Appendix A, Figures A-8 and A-9 respectively.

We use a temporary user model in the adaptation module for creating and maintaining the user model in RDF. The user profile is continuously changing. For instance users can change, delete or add information items. These changes are managed by the adaptation module, which allows additions, deletions and changes using a user ontology, which is an ontology developed to represent user profiles within the ECS CMWP.

The user interface of SEMPort is implemented using a set of Java Servlets on the Web server (i.e. Tomcat), through which users can communicate with the system. When a user makes a request to the Web server, the corresponding module which is a Servlet queries the KB over HTTP using SPARQL queries. Users can access SEMPort's Web interface from an ordinary Web browser and can communicate with the portal through the functionalities of SEMPort, such as semantic navigation with personalized views, ontology-based search and personalized homepages.

6.2.1 Reuseability of SEMPort in Different Ontology Domains

To enable re-usability, SEMPort is implemented generically with re-usable components. For instance, different ontology files can be uploaded to the KB, using Protégé or a Web front-end. The content editing Web interface is generic and can be used for the maintenance of different instances of different ontologies.

Semantic navigation and ontology-based search can be used for browsing different ontologies. Since the personalization is typically application specific, it is adapted to our testing domain, ECS CMWP for the purpose of illustration. A user ontology is used to adapt the contents to the needs of the users. However, personalization can be adapted to different domains by changing the user ontology. Also, context-based semantic links are generated using the underlying ontologies and rules. Explicit and inverse links are generic and can be generated on any domain without modifications. Implicit links and recommendation links can be tested on different domains, by changing the rules. Additionally, SEMPort is accessible from different graphical user interface browsers, since it is implemented in a browser independent manner. For instance, SEMPort has been successfully tested on the Internet Explorer browser, the Mozilla FireFox browser and the Netscape Navigator browser (see Figures A-3 and A-4 in the Appendix A). In the remainder of this report, the functionalities of SEMPort are illustrated using the Internet Explorer browser.

6.2.2 Implementation

The Web interface of SEMPort is implemented using Java Servlets and the Jena Ontology Framework. Java Servlets are very convenient for serving dynamic Web content, session tracking and database connectivity using JDBC. JDK1.5 was used for the implementation of SEMPort. As a Web server, Apache Tomcat 5.5 was used on port 7070. For creating and handling semantic metadata, we have utilized Jena, which is a Java framework for building Semantic Web applications. It provides a program environment for RDF, RDFS, OWL and SPARQL and includes a rule-based inference engine. In SEMPort, Jena 2.4 was used. In addition, the Jena rule-based reasoner and the Jena OWL Micro reasoner (Jena Inference, 2008) were utilized for supporting inferencing. ARQ is a query engine for Jena, which supports the SPARQL query language, RDQL and internal query language ARQ. In SEMPort, ARQ 2.5 was used for SPARQL querying. ARQ is compatible with the latest SPARQL developments. MySQL 4.1 was used as a back-end storage for Jena models, which is also known as the KB. MySQLwas connected to the Tomcat Web Server and the Jena Ontology Framework using a JDBC driver (Java MySQL connector 3.1.12). For the maintenance of ontologies and instances, the Protégé Ontology API was connected to the MySQL using the Protege2Jena plug-in version 3.2 (Ptotege2Jena, 2008). It is possible to import/export ontologies from/to MySQL and Protégé.

6.3 Demonstration of SEMPort on the ECS CMWP

For the demonstration of the proposed semantic portal, we decided to use ECS CMWP. The ECS CMWP consists of a list of modules which are categorized into different degree courses, e.g. COMP for Computer Science. The information about each module is presented in a syllabus information page, which shows the module's credit value, name, exam percentage, prerequisites, etc. (see Figure A-1 in Appendix A). The page also shows a list of topics which are covered by the module. However, in the module's web page, hyperlinks between information items are often disconnected, relevant and interconnected pages are not provided and there is no search mechanism. To resolve these problems, we decided to use the ECS CMWP for an experimental implementation of SEMPort. In order to use the ECS CMWP, first we needed to extract metadata from Web pages and annotate it with metadata using ontologies.

In the remainder of this section, first we explain how we annotated ECS CMWP with domain ontologies and then the proposed user ontology is discussed. In the rest of the thesis, we use the ECS CMWP domain for illustrating the functionalities of SEMPort.

6.3.1 Annotating ECS CMWP with Domain Ontologies

In a semantic portal, the ontology is the backbone of the architecture. It classifies various entities, associates relationships between entities, and more importantly, allows us to infer implicit knowledge. Since the knowledge is represented by semantic structures, the portal's contents can be accessed by both humans and software agents. Also, the contents can be easily maintained by different users or software agents conforming to ontologies. Additionally, it is possible to reason about the knowledge and performing personalization.

For the annotation of the ECS CMWP, we used two existing ontologies: a part of the ECS Ontology (ECS Ontology, 2006), which we call ECS_COURSE, and the ACM Computer Classification System (ACM CCS, 2008).

6.3.1.1 ECS_COURSE

ECS_COURSE is used to annotate the ECS CMWP. The main entity types of this ontology are: Agent, Cohort, Degree, Individual, Module, Time Entry, and Year of Study. Each of these entity types is divided into more narrowly defined entities and each entity can participate in relationships with other entities.

ECS_COURSE and its instances are created manually in the OWL DL, by using the Protégé Ontology Editor. The visualization of ECS_COURSE is presented using OWL Viz[45] as shown in Figure A-2 in the Appendix A. SEMPort can handle ontologies that are written in OWL Lite or OWL DL. In addition, for the purpose of readable

[45] www.co-ode.org/downloads/owlviz/co-ode-index.php [last accessed, 22/12/2008]

presentation of the ontology and the instances, we annotate entities and attributes with the "rdfs[46]:label" property, and instances with the "dc[47]:title" property. At run-time, if these properties are empty, the URI of the entity, attribute or instance is used, and the prefix of the URI is removed from the presentation.

6.3.1.2 ACM Computer Classification System

The ACM Computer Classification System (ACM CCS, 2008) has been used to classify scientific publications in the field of computer science for many decades. We used this to annotate topics covered by the ECS course modules. The ACM CCS is made up of eleven main categorizations, which are further divided into more specific levels. Part of this classification hierarchy is reproduced in Figure 6-4. Classification numbers in parentheses (e.g. J.7 in Figure 6-4), are used to relate to different concepts in the ACM CCS.

- General Literature
- B. Hardware
- C. Computer Systems Organization
    - C.1 PROCESSOR ARCHITECTURES
    - C.2 COMPUTER-COMMUNICATION NETWORKS
    - C.3 SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS (J.7)
        - Microprocessor/microcomputer applications
        - Process control systems
        - Real-time and embedded systems
        - Signal processing systems
        - Smartcards
    - C.4 PERFORMANCE OF SYSTEMS
    - C.5 COMPUTER SYSTEM IMPLEMENTATION
    - C.m MISCALLENEOUS
- D. Software
- ….
- K. Computing Milieux

Figure 6-4 Part of the ACM CCS

In order to make use of this classification, first we needed to convert it to RDF and then combine it with ECS_COURSE. The ACM CCS was converted to RDF using Simple Knowledge Organization System (SKOS) Vocabulary (SKOS, 2008). SKOS is a simple knowledge representation that is used to specify knowledge organization systems, such as thesauri, classification schemes and taxonomies within the framework

[46] RDFS namespace, http://www.w3.org/2000/01/rdf-schema# [last accessed, 22/12/2008]

[47] Dublin Core namespace, http://purl.org/dc/elements/1.1/ [last accessed, 22/12/2008]

of the Semantic Web. Then, we extended ECS_COURSE, so that ECS_COURSE instances could be linked to ACM CCS instances.

The ACM CCS classification is available from the ACM domain. We used the ACM CCS ascii file[48] and converted it to RDF by using Perl Scripts. Every classification in ACM CCS (i.e. B. Hardware in Figure 6-4) and every ACM CCS topic (i.e. Smartcards in Figure 6-4) is annotated as an instance of the “skos[49]:Concept” entity. We used our own namespace while naming ACM CCS instances (i.e. http://localhost:7070/SEMPort/modules.owl#). In addition, the “skos:broader” relationship was used to create sub-area relations between different ACM CCS topics. To label ACM CCS topics, we used the “skos:prefLabel” property. To annotate explanations that are defined in the ACM CCS, the “skos:description” attribute was used. Also in the ACM CCS taxonomy, different topics are related to each other by providing the topics’ classification number in parenthesis. For example in Figure 6-4, C.3 is related to J.7 by providing its classification number in parenthesis. To get the benefit of this kind of relationship, the “skos:related” relationship was used. A part of the classification represented in Figure 6-5 is annotated as follows according to the SKOS vocabulary.

```
<rdf:RDF
xmlns:skos="http://www.w3.org/2004/02/skos/core#">
<skos:Concept rdf:about="http://localhost:7070/SEMPort/modules.owl#C.3-Smardcards">
    <skos:prefLabel>Smartcards</skos:prefLabel>
    <skos:broader rdf:resource="http://localhost:7070/SEMPort/modules.owl#C.3"/>
</skos:Concept>
<skos:Concept rdf:about="http://localhost:7070/SEMPort/modules.owl#C.3">
    <skos:prefLabel>SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS</skos:prefLabel>
    <skos:broader rdf:resource="http://localhost:7070/SEMPort/modules.owl#C"/>
    <skos:related rdf:resource="http://localhost:7070/SEMPort/modules.owl#J.7"/>
</skos:Concept>
</rdf:RDF>
```

Figure 6-5 A part of ACM CCS annotated by SKOS vocabulary

In the context of ECS CMWP, this classification is a very convenient one with which to annotate topics covered by the ECS modules. To classify a module with ACM CCS, we added a new attribute “hasTopic” to ECS_COURSE. For instance, a module may teach a number of ACM CCS topics and this relationship is defined by “hasTopic” attribute as shown below:

48 http://www.acm.org/about/class/1998/ [last accessed, 12/11/2008]

49 http://www.w3.org/2004/02/skos/core [last accessed, 5/1/2009]

```
<rdf:RDF xmlns:portal="http://localhost:7070/SEMPort/modules.owl#">
<portal:ModuleInSession rdf:about="http://localhost:7070/SEMPort/modules.owl#COMP1002">
    <portal:hasTopic rdf:resource="http://localhost:7070/SEMPort/modules.owl#C.3"/>
</portal:ModuleInSession>
</rdf:RDF>
```

Figure 6-6 An annotated ECS course module with SKOS classification using hasTopic attribute

6.3.2 User Ontology for the ECS CMWP

The user model represents relevant user characteristics, such as background, interests or preferences. The majority of AH systems use an *overlay model* of user knowledge. (Brusilovsky, 2003) states that "*The key principle of the overlay model is that for each domain model concept, an individual user knowledge model stores certain data that is an estimation of the user knowledge level on this concept.*" He suggests an alternative is the *historic model* that keeps some information about user visits to individual pages. In addition, characteristics of a user might be determined by modeling groups of users with similar requirements. Information about user groups can be available either as *stereotypes* or dynamically calculated as user *group models* (Kobsa, 2001).

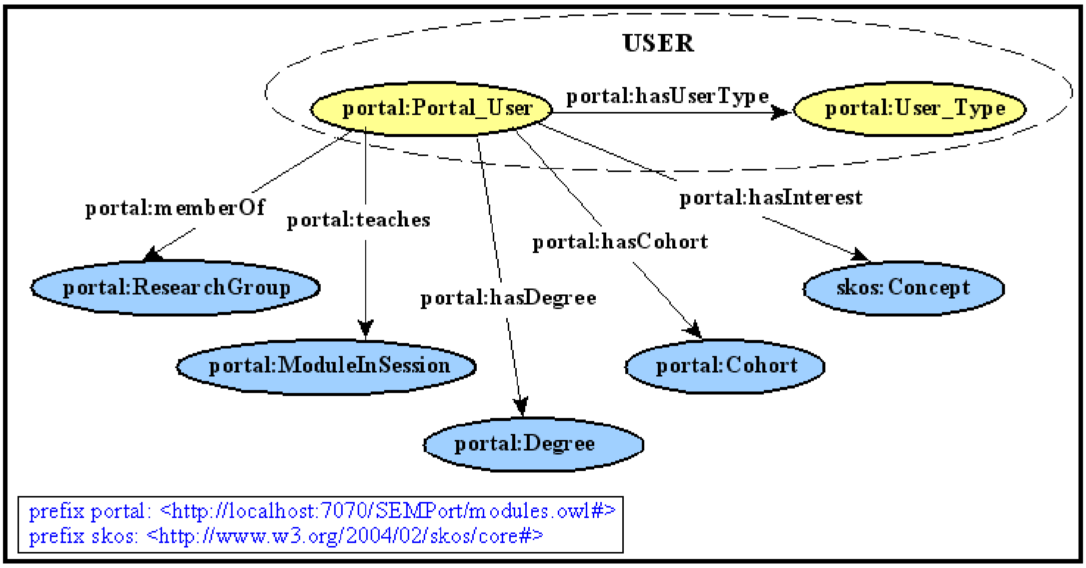


Figure 6-7 Relationships between USER and ECS_COURSE

In order to support personalization in the ECS CMWP application, we need to take into account users of this domain. The users of the ECS CMWP are undergraduate students, postgraduate students, lecturers and other staff such as administrators. These users have different needs when they use the ECS Website. In order to adapt to these needs, we generated a user ontology, which is called USER. USER is used to model user background and interests, as a set of concepts that have relationships to domain model concepts, and is very similar to the overlay model. We used these relationships for the purpose of personalization. The relationship between USER and ECS_COURSE is presented in Figure 6-7.

6.3.2.1 Concepts Used in the USER

USER is a very simple ontology that has two entities: Portal_User and User_Type. The User_Type entity is used to classify different portal users, such as student, lecturer and other. Each user is identified as an instance of the Portal_User entity. The Portal_User concept is used to create relationships between the user and the domain model concepts as shown in Figure 6-7. Hence, depending on different user types, diverse relationships exist between the user and the domain entities. For example, users can add relevant ACM CCS topics into their profiles and provide different weights depending on their interests, such as low, medium and high.

6.3.2.2 Set of Properties Used in the USER Ontology

Table 6.1 summaries the set of properties used to connect the concepts used in USER to the domain ontology concepts.

Table 6-1: Properties of the USER

Property Name	**Definition**
hasUserType	Defines the type of the user. In the USER, we have three predefined instances: student, teacher and other. hasUserType can take one of these instances values.
About	Defines the URI of the user's interest. It can take values of the ACM CCS instances (i.e. http://localhost:7070/SEMPort/modules.owl#C.3)
hasWeight	Defines the weight (i.e. importance) of the interest. In the USER, we have three predefined instances: low, medium and high. hasWeight can take values from these instances.
hasDegree	For students only. Defines the degree of the student. Domain of this property is Portal_User entity from the USER and it takes values

	from Degree entity of the ECS_COURSE.
hasCohortOf	For students only. Defines the cohort of the student. Domain of this property is Portal_User entity from the USER and it takes values from Cohort entity of the ECS_COURSE.
hasYear	For students only. Defines the current year of study of the student. Domain of this property is Portal_User entity from the USER and it takes values from Year of Study entity of the ECS_COURSE (1-6).
teaches	For teachers only. It assigns a set of teaching modules to the user from the ModuleInSession entity of the ECS_COURSE.
isMemberOf	Contains research group(s) that the user is member of. Domain of this property is Portal_User entity from the USER and it takes values from the Research Group entity of the ECS_COURSE.

6.3.2.3 The Semantic Metadata of a User Profile

An example user profile using USER is represented in Figure 6-8.

```
<rdf:RDF
      xmlns:portal="http://localhost:7070/SEMPort/modules.owl#"
      xmlns:skos="http://www.w3.org/2004/02/skos/core#">
<portal:Portal_User rdf:about="http://localhost:7070/SEMPort/modules.owl#jn03">
   <portal:hasUserType rdf:resource="http://localhost:7070/SEMPort/modules.owl#student"/>
   <portal:hasCohortOf rdf:resource="http://localhost:7070/SEMPort/modules.owl#cs_3"/>
   <portal:hasDegree rdf:resource="http://localhost:7070/SEMPort/modules.owl#csBSc"/>
   <portal:hasInterest>
      <portal:About>http://localhost:7070/SEMPort/modules.owl#H.5.2-graphical-user-
      interfaces</portal:About>
      <portal:hasWeight>1</portal:hasWeight>
   </portal:hasInterest>
</portal:Portal_User>
</rdf:RDF>
```

Figure 6-8 An example user profile using USER is represented in RDF/XML syntax

6.4 Maintenance of the Portal Contents

In SEMPort, three different maintenance mechanisms are supported for keeping the portal’s content up-to-date, which is very important for the development cycle of a semantic portal: the Protégé ontology editor is connected to the KB, a Web-front end is developed to upload ontology/instance files and a Web-interface is implemented for editing existing metadata. In Figure 6-9, the SEMPort maintenance mechanisms are shown.

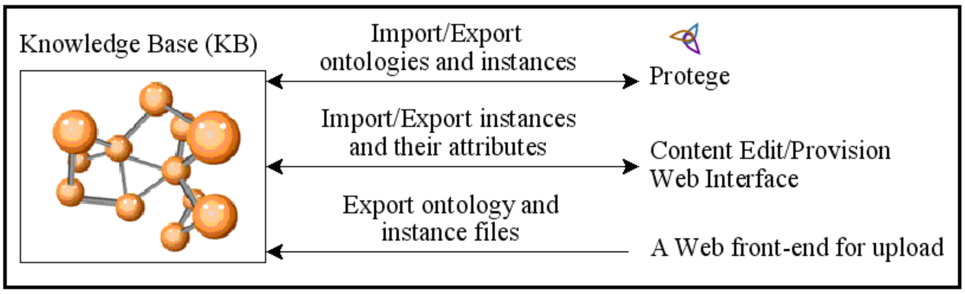


Figure 6-9 Maintenance mechanisms for SEMPort

6.4.1 Connecting Protégé Ontology Editor to the KB

Protégé is used for maintenance purposes, and it enables export or import of the ontologies to or from the KB. An administrator can import ontologies and their instances from the KB to the Protégé and can make changes to both the ontology and the instances. Changes can be saved to the KB by overwriting the existing information or by cleaning the domain models before an export as shown in Figure 6-10. Any valid OWL Lite or OWL DL ontology files can be used and changes to the KB can be viewed at run-time without restarting the Web server.

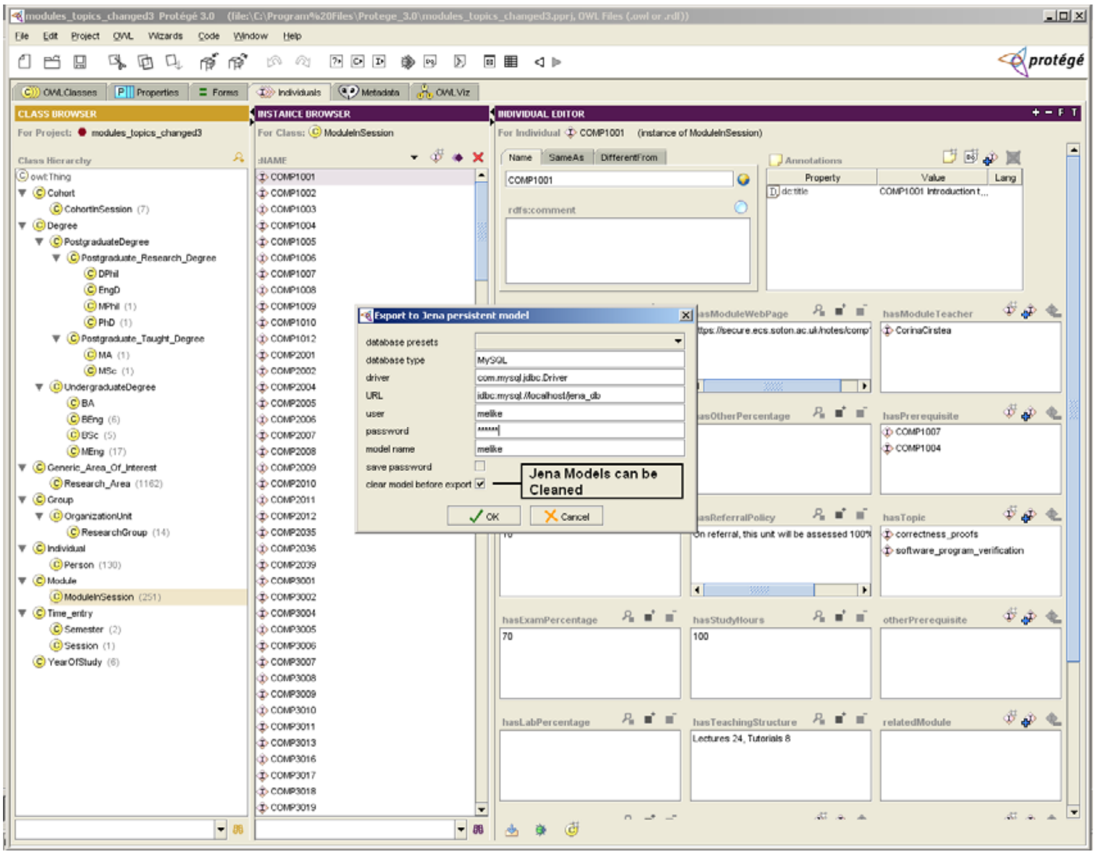


Figure 6-10 The export of the ontology and their instances from Protégé to the KB using ProtégétoJena plug-in

6.4.2 A Web-Front End for Uploading Metadata to the KB

Alternatively, a Web front-end is generated to allow the upload of any number of ontology and instance files to the KB (Figure 6-11). From the Web front-end, an administrator can clean the domain models before an upload or can aggregate newly-added ontology files with the existing ontologies in the KB.

The Web front-end also provides consistency checking on the ontologies and the instances. Validation of the KB is tested by the Jena OWL reasoner. After the KB is updated, we perform consistency checking on the concepts and their instances, and property domain and range values. Then, a validation report is displayed that contains any warnings or errors. Any valid OWL Lite or OWL DL ontology files can be used and changes to the KB can be viewed at run-time without restarting the Web server.

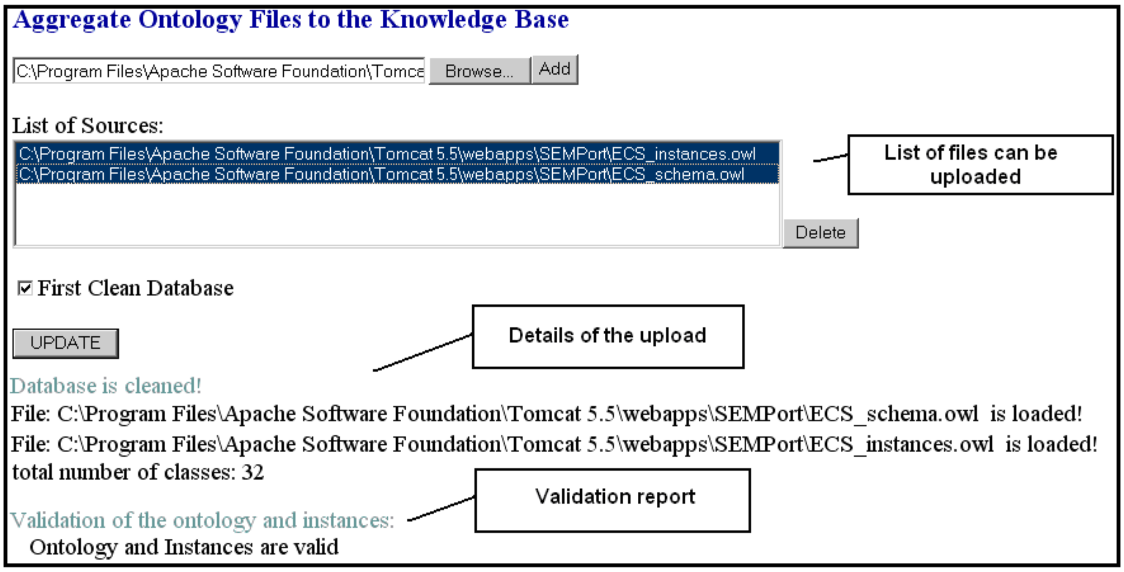


Figure 6-11 A Web front-end for uploading ontology files to the KB

6.4.3 Real-time Content Edition Provision Web Interface

SEMPort also supports real-time content editing and provision through its Web interface. Authorized users can change or add information using a distributed Web interface. This interface will be explained in more detail later in this chapter (section 6.10).

6.5 Semantic Navigation

The basic aims of semantic navigation in SEMPort are to provide clearly organized and easily traversable presentation of all the contents in the portal. In SEMPort, the ontology is used to supply browsing. For example, the left pane in Figure 6-1 shows the ontology hierarchy and provides links to different ontology concepts. The numbers in parentheses indicate how many instances will be found. Users can narrow or widen concepts by clicking the "+" or "–" buttons, and can also see all concepts of the hierarchy, by clicking on the "show all" button as shown in Figure 6-12.

In SEMPort, there are two kinds of navigation views; a general view (Figure 6-1, right pane) and a detailed view (Figure 6-2). We used a simple approach for visualization. More complex visualization mechanisms and HCI issues (usability of presentation or visualization) are not in the scope of this thesis.

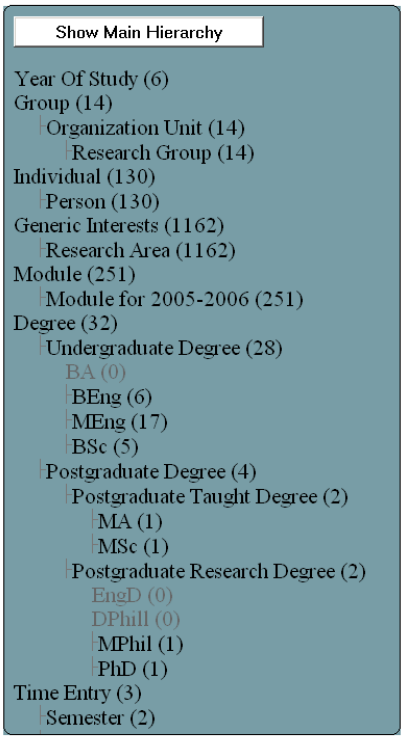


Figure 6-12 The whole ontology hierarchy can be presented during semantic navigation

6.5.1 General View

The general view is activated when an instance is selected from the ontology hierarchy. The instances of the selected concept are then presented on the right pane as illustrated in Figure 6-1. In this view, only instance titles are shown. Note that instances are also sorted (link sorting) based on the similarity of a link to the user's interests, if the user is logged in to the portal. However, if the user does not log in, then instances are presented in alphabetical order. In addition, the content is divided into pages and users can arrange the number of items per page from general view interface. Additionally, in the general view, the user can perform context-specific searches using the ontology-based search. Given that, an automatic ontology-based search form is generated depending on the selected concept type and presented to the user in the general view.

6.5.2 Detailed View

When a user clicks onto an instance from the general view, a detailed view is opened as seen in Figure 6-2. The detailed view shows all information about a particular instance (i.e. all attributes and their values), as well as displaying semantic hyperlinks, such as links to directly associated resources, inverse links, implicit links and recommendation links. These links are explained in further detail later.

6.6 Ontology-Based Search

Ontology-based search provides detailed querying of a certain concept using domain-specific attributes. In the OntoWeb portal (Spyns et al., 2002), ODESeW (Corcho et al., 2003) and REASE (REASE, 2008), ontology-based search is used as a separate mechanism to query the KB. However, in our approach, to assist the user in locating certain information, ontology-based search is integrated into the semantic navigation (Figure 6-2). Hence the user can narrow the information space by performing concept-specific searches during the browsing. In addition, entering search values to relationships are difficult in the OntoWeb portal and ODESeW. To solve this problem in SEMPort, valid search values for relation properties are automatically provided by the ontology-based search. In the Figure 6-13, examples of automatically provided search values during the ontology-based search are shown.

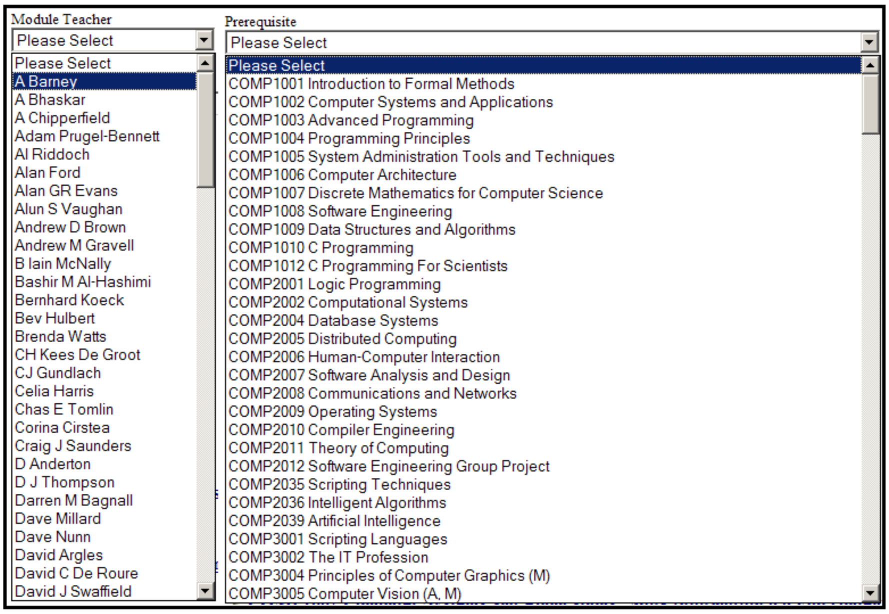


Figure 6-13 Examples of automatically provided relationship values generated during an ontology-based search

Ontology-based search is triggered when the user selects a concept from the ontology hierarchy. Depending upon the selected concept, a dynamic search form is automatically generated by the ontology-based search module. All the properties of this concept and its super-concept properties are displayed. If the value of the property is a literal, a text box is shown to fill in keywords, or if the value is a relationship, a drop-down list is displayed. The instances and the subclass instances of the property's range-class are filled alphabetically into the drop-down list; thus the user is guided during the search. On numeric values, the user can make restrictions, such as equivalence (=), greater (>), (<), etc. Users can enter search values to one or more attributes at the same time. Search results will contain the conjunction of the entered data.

Figure 6-14 illustrates an ontology-based search on the "Modules for 2006-2007" concept. The user is looking for course modules, which are taught in "Semester 1", teaches "software engineering" and has "Coursework percentage" greater or equal to 50. She selected "semester 1" for "Taught In Semester" attribute, selected "SOFTWARE ENGINEERING" for "Teaching Topic" attribute and entered ">=50" for "Coursework Percentage" attribute. Note that the other attributes have remained blank. Then, a search is performed on the "Modules for 2006-2007" concept, and results are displayed as a general view as shown in the right pane at Figure 6-14.

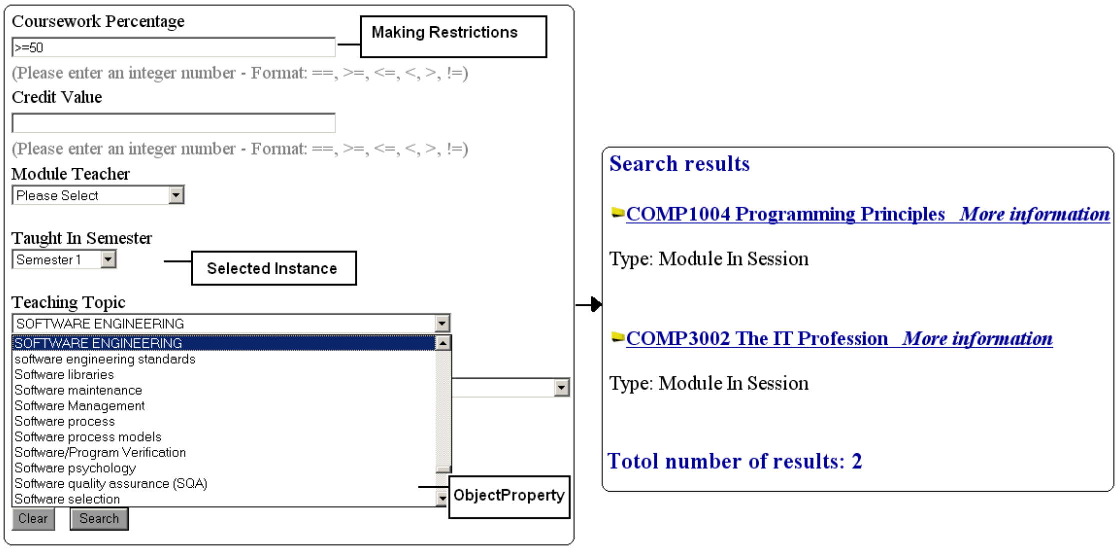


Figure 6-14 Ontology-based search example

At anytime during browsing, the user can launch a search by entering some information into the search form. The search form actually contains the URI of the properties and the search values to those attributes. When a search form is submitted, the ontology-based search module converts the entered search values into a SPARQL query using the URIs of the attributes, URIs of the properties and provided search values. If the entered value is string or numeric, then, SPARQL filter statements are also generated. For example, the search that is illustrated in Figure 6-14 is converted to the following SPARQL query. For simplicity we used prefixes for this query.

```
PREFIX portal: <http://localhost:7070/SEMPort/modules.owl>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT * WHERE {
?x portal:taughtInSemester portal:semester_1.
?x portal:hasTopic portal:software_engineering.
?x portal:hasCourseworkPercentage ?percentage.
FILTER (?percentage>=50).    }
```

Figure 6-15 Created SPARQL search query

The KB is interrogated for matching instances using the search SPARQL query. The search is executed on the inferred knowledge, which means sub-concepts are also interrogated for matching instances. To preserve consistency throughout the presentations, search results are presented as general views. When a user clicks onto an instance from the general view of the search results, a detailed view is opened, which shows all the information about a particular instance.

6.7 Inferencing

In a semantic portal, it is possible to perform reasoning on the metadata of the portal's contents and implicit information can be presented to the user. This ability is one of the greatest benefits of utilizing the Semantic Web technologies in semantic portals. Thus, in SEMPort we utilize inferencing using the inference module. The inference module is responsible from inferring implicit links between domain instances and presenting them to the user. For instance, inferred links can be shown in the search results or in the detailed view of a semantic instance (i.e. implicit links and recommendation links). The inferred links are generated from a set of inference rules to the inference engine by a separate file.

6.7.1 Rule Syntax and Reasoning Pipeline

In SEMPort, a Jena rule-based reasoner is utilized for reasoning (Jena Inference, 2008). Rules are encoded in datalog syntax and they are executed in top-to-bottom, left-to-right order with backtracking. The rules will only function if there is matching semantic descriptors during the SPARQL querying. Figure 6-16 shows the interaction of reasoning rules pipeline. We have chosen to use backward chaining rules, because backward chaining rules (Jena Inference, 2008) are executed faster compared to forward chaining rules in our case study. In forward chaining, when a new query is executed on the reasoning engine, all rules in the rule-base are tested and fired. Then results are stored for future matches. In a big dynamic KB, many different semantic descriptors can be queried from the reasoning engine and each time all rules are tested and fired, which can be a time-consuming task. On the other hand, in backward chaining, only matching rule heads are fired and stored, so that previous matches to a goal are recorded and used when satisfying similar future goals. This provides quick resoning on a dynamic KB.

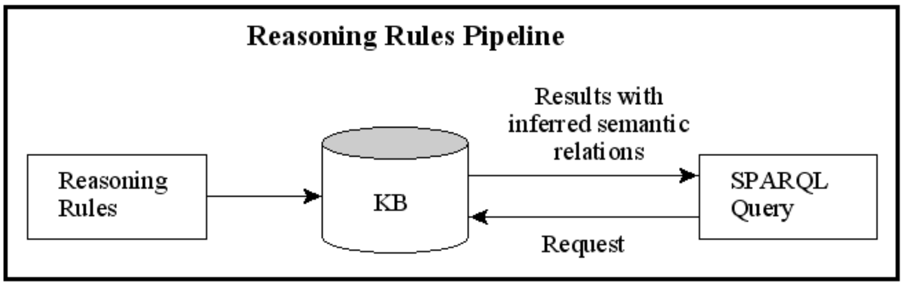


Figure 6-16 Reasoning rules pipeline

In Figure 6-17, an example inference rule is presented which is used to reason about the module teachers. Assume, the user wants to learn about all modules taught by a specific teacher and performs an ontology-based search on the ModuleInSession concept. She selects the name of the teacher from the drop-down list of the "portal:ModuleTeacher" property and performs a search. In the search result, she assumes all modules that are taught by this teacher will be given. However, in ECS_COURSE, the module teacher and the module leaders are separated. A search on the module teacher, may not give all modules taught by a teacher, if the teacher is the module leader of a course. To solve this problem, inference rules can be used to state that the module leader is also the module teacher as shown in Figure 6-17. Thus, the search results will contain all modules taught by a specific teacher.

```
[Teacher: (?module portal:hasModuleTeacher ?leader)
<- (?module portal:hasModuleLeader ?leader)]
```

Figure 6-17 Reasoning rule excerpt for the hasModuleTeacher descriptor

In SEMPort, reasoning rules are also utilized to generate implicit and recommendation links during the semantic navigation. These rules will be explained in detail in the next section.

6.8 Context-Based Automatic Link Addition

In SEMPort, four different kinds of links, which we call semantic hyperlinks are generated depending upon the context: explicit links, inverse links, implicit links, and recommendation links. These links can be categorized into two groups: hand-made links and computed links. According to (Ashman et al., 1997), hand-made links are created by a human and computed links are automatically created by a computation. In addition, if the same set of computed links is generated by the same computation, then they are referred to as functional links (Ashman and Verbyla, 1993). In other words, functional links are sets of hyperlinks that are dynamically computed from the same computation. According to this classification, explicit links can be categorized as hand-made links, since they are generated by a human using the underlying ontologies. Inverse, implicit and recommendation links are different types of functional links which are dynamically computed by using different computations.

The main aim of the semantic hyperlinks is to improve user navigation by using automatically generated context-based links. These links are presented during the detailed view of a particular instance. In this way, links to relevant pages are provided during navigation. Semantic hyperlinks are generated using a combination of rules and SPARQL queries that are explained in more detail in the following sub-sections. These rules are created by using the Jena rule-based reasoner and reasoning rules and executed by the inference module. Also, a link represents the URI of the instance.

6.8.1 Related Work

COHSE (Carr et al., 2001) and Magpie (Dzbor et al., 2003; Domingue et al., 2004; Dzbor et al., 2007) are two systems that are used to provide extensive linking; they provide links to related resources based on the underlying ontology, using reasoning and services. However, extensive linking is not provided in most of the semantic portals that were discussed in the literature review. Only MuseumFinland (Hyvonen et al., 2004; Hyvonen et al., 2004b) gives importance to recommendation links for improving browsing by users. These links are generated using rules. In MuseumFinland, reasoning is supported by a separate server, and recommendation links are generated by querying this server from the multi-facet search engine. However, in SEMPort, recommendation links are automatically generated, because during the querying of the KB, rules are automatically inherited.

6.8.2 Explicit Links

Explicit links are used to provide links to directly associated resources using the ontologies. Explicit links are created manually during the generation of ontologies/instances and they provide relationships between different instances. Assume the KB contains a number of instances, C_1, C_2, …, C_n (where n corresponds to the number of instances in the KB) and predicates are used for defining relations between instances. If there is a relationship between two instances, the relationship can be shown as a finite set of First Order Logic (FOL) sentence:

$$\text{relation}(C_i, C_j), \text{ for certain } C_i \neq C_j \qquad (6.1)$$

During the semantic navigation, if an instance has a relation to another instance as described above, then the connected instance is shown as an explicit hyperlink. These hyperlinks are generated by using SPARQL queries; if the relation type is an ObjectProperty, then its value is presented as a link. For example, in Figure 6-2, “Module Leader” and “Module Teacher” are relationships which are defined by the ontology, and their values are shown as explicit hyperlinks.

6.8.3 Functional Links

Inverse, implicit and recommendation links are various sets of hyperlinks, which are dynamically generated from different computations. Therefore, they can be referred to as functional links. In this section, we explain functional links in more detail.

6.8.3.1 Inverse Links

Inverse links are types of functional links that are dynamically computed during the presentation of a page. Inverse links allow information resources to be accessed in backward direction. Inverse links are generated by using the relation between two concepts; explicit links are presented as inverse links at the connected instance. Again, these links are generated by using SPARQL queries. During the detailed view of an instance, the KB is interrogated for relations that have the current instance as an object in a (subject, predicate, object) triple. Then, the matching relationships are presented as inverse links at the current instance’s detailed view.

In Figure 6-2, “is Module (Compulsory) of” provides an inverse link to “second year Computer Science and Software Engineering” cohort, which means that COMP2012 is a compulsory module for that cohort. Also, when the user follows the link to the “Project and People Management” topic from Figure 6-2, she can see all modules which include the particular topic by using the inverse links, such as COMP2012 and COMP3002 presented in Figure 6-18.

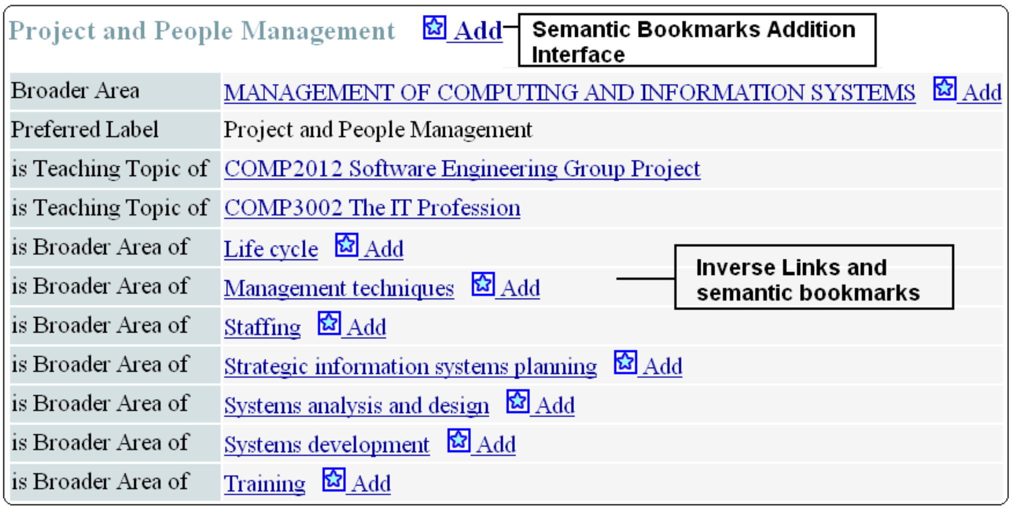


Figure 6-18 Illustration of inverse links and bookmark addition interface

6.8.3.2 Implicit Links

Implicit links are types of functional links that are dynamically computed using a set of inference rules during the presentation of a page. The KB contains indirect relations, which can be used to support the navigation of the user. Therefore, the purpose of the implicit links is to infer indirect relations from the KB and present it to the user.

In SEMPort, prerequisites of prerequisites are used to present implicit hyperlinks. However, in different domains, diverse implicit relationships can be used to provide implicit links. Assume the KB contains the list of course modules, M_1, M_2, …, M_i (where i corresponds to the number of modules in the KB). A module may have a prerequisite, which is defined by the ontology and can be shown with FOL as follows:

$$\text{preq}(M_j, M_k), \text{ for certain } M_j \neq M_k \tag{6.2}$$

A prerequisite module may have another prerequisite, and indirectly the prerequisite of a prerequisite becomes a prerequisite to the module. Therefore, during the presentation of an instance, if the instance type is a “Module”, links to a prerequisite of prerequisites can be inherited by using the following FOL:

$$\begin{gathered}\forall M_j \forall M_k (\text{preq}(M_j, M_k)) \wedge \exists M_z (\text{preq}(M_k, M_z)) \wedge M_j \neq M_k \wedge M_k \neq M_z \\ \Rightarrow \text{preq}(M_j, M_z)\end{gathered} \tag{6.3}$$

The FOL rule explained for prerequisites of prerequisites in equation (6.3) is implemented in SEMPort using the following Jena backward reasoning rule as shown in Figure 6-19. In this example, we created a new relationship property, “portal:otherPrerequisite” for this newly inferred relation. Diverse relations between ontology instances can be used to generate implicit links.

```
@prefix portal: http://localhost:7070/SEMPort/modules.owl #namespace
[Prerequisite: (?module portal:otherPrerequisite ?pre2)
<-
(?module portal:hasPrerequisite ?pre1),
(?pre1 portal:hasPrerequisite ?pre2),
notEqual(?pre1,?pre2), notEqual(?module,?pre2)]
```

Figure 6-19 Reasoning rule excerpt for prerequisites of prerequisites

For example, in Figure 6-2, COMP2012, has prerequisite COMP2007, which is explicitly defined by the ontology. In addition, COMP2007 has other prerequisites, COMP1003 and COMP1008, which means indirectly COMP2012 is restricted to those prerequisites. Therefore inherited knowledge is used to create implicit links and presented to the user as shown in Figure 6-2 (i.e. Prerequisite other property).

6.8.3.3 Recommendation Links

Recommendation links are types of functional links that are dynamically computed using a set of rules during the presentation of a page. Implicit and recommendation links are different from each other, since they use different sets of rules for the computation and the created links generate different relationships between instances.

In SEMPort, recommendation links are used to provide interesting relationships between different concepts and assist users in finding related information. Therefore, recommendation links suggest related pages based on the context. In the ECS CMWP domain, ACM CCS topics covered by the modules are used to present related topics as can be seen in Figure 6-2 (i.e. presented links under related topics). Assume the KB contains the list of course modules, M_1, M_2, …, M_i (where i corresponds to the number of modules in the KB) and a module may cover the list of ACM CCS topics, T_1, T_2, …, T_j (where j corresponds to the number of topics in the KB). The topics may have super-area topics, such as described by skos:broader relationship. The topics may have relations to other topics, such as stated by using skos:related relationship. These relationships are described in more detail in section 6.2.1.2 and are represented in the following FOL sentences:

$$
\begin{aligned}
&\text{@prefix portal}:\text{http}://\text{localhost}:7070/\text{SEMPort/modules.owl\#}\\
&\text{@prefix skos}:\text{http}://\text{www.w3.org/2004/02/skos/core\#}\\
&\text{@prefix rdfs}:\text{http}://\text{www.w3.org/2000/01/rdf-schema\#}\\
&\exists M_k\ \exists T_l,\ \text{portal}:\text{hasTopic}(M_k,\ T_l),\ M_k \text{ is a Module and } T_l \text{ is a ACM CCS Topic}\\
&\exists T_b,\ \text{skos}:\text{broader}(T_l,\ T_b)\\
&\exists T_r,\ \text{skos}:\text{related}(T_l,\ T_r)
\end{aligned}
\tag{6.4}
$$

During the presentation of an instance, when the instance type is a "Module", the topics covered by the module are used to generate recommendation links by using the following FOL rules described in equation (6.5). The rdfs:seeAlso property is used to create recommendation links to other related instances.

$$
\begin{aligned}
&\forall \mathrm{M}_k\ \forall \mathrm{T}_l(\text{portal}:\text{hasTopic}(\mathrm{M}_k,\mathrm{T}_l)) \wedge \exists \mathrm{T}_b(\text{skos}:\text{broader}(\mathrm{T}_l,\mathrm{T}_b)) \wedge \mathrm{T}_l \neq \mathrm{T}_b\\
&\qquad\qquad \Rightarrow \text{rdfs}:\text{seeAlso}(\mathrm{M}_k,\mathrm{T}_b)\\
&\forall \mathrm{M}_k\ \forall \mathrm{T}_l(\text{portal}:\text{hasTopic}(\mathrm{M}_k,\mathrm{T}_l)) \wedge \exists \mathrm{T}_r(\text{skos}:\text{related}(\mathrm{T}_l,\mathrm{T}_r)) \wedge \mathrm{T}_l \neq \mathrm{T}_r\\
&\qquad\qquad \Rightarrow \text{rdfs}:\text{seeAlso}(\mathrm{M}_k,\mathrm{T}_r)
\end{aligned}
\tag{6.5}
$$

Based on the relationships between topics, which is defined above, recommendation links to the broader ACM CSS topics and recommendation links to the related ACM CCS topics are generated by the two reasoning rules described in Figure 6-20. The first rule provides hyperlinks to more general topics, and the second rule uses the relations between taxonomies to supply interrelated topics. Links to specific topics can also be shown. In Figure 6-2, related topics are recommended at the bottom of the Web page using the reasoning rules described in Figure 6-20.

```
@prefix portal: http://localhost:7070/SEMPort/modules.owl#
@prefix skos: http://www.w3.org/2004/02/skos/core#
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema#

[Hierarchy: (?module rdfs:seeAlso ?topic2)
<-
(?module portal:hasTopic ?topic1),
(?topic1 skos:broader ?topic2),
notEqual(?topic1,?topic2) ]

[Relations: (?module rdfs:seeAlso ?topic2)
<-
(?module portal:hasTopic ?topic1),
(?topic1 skos:related ?topic2),
notEqual(?topic1,?topic2) ]
```

Figure 6-20 Reasoning rule excerpt for the ACM CCS recommendation links

6.8.3.4 Notes on Context-Based Hyperlinks

In SEMPort, the OWL reasoner and the generic rule-based reasoner are connected to the KB and the presentation is generated by querying the KB with SPARQL queries. During querying, if the SPARQL query matches the head of a rule (goal), then the rule fires and inherited knowledge is deduced. In this way, semantic hyperlinks are automatically generated based on the context and presented without any other processing. In addition, to adapt semantic hyperlinks to different ontology domains, we only need to change the rules. Thus, semantic hyperlinks can be easily adapted to diverse domains. Figure A-9 in the Appendix A shows an example of rules that were used in the ECS CMWP case study. Also, different kinds of reasoners can be bound to the KB as shown in Figure A-8.

6.9 User Modelling

In order to support personalization, SEMPort uses the USER ontology for representing the interests and background of users. In this section, we discuss how the user profiles are generated and maintained by SEMPort.

6.9.1 Registering into SEMPort

In SEMPort, in order to start personalization, first the user has to login to SEMPort with their ID and password. The first time a user logs in, certain questions are asked to ascertain some background (such as name, occupation, degree, etc.) and the ACM CCS topics they are interested in. Depending on the different user types (student, teacher, and other), different questions are asked (see Figure A-10 in the Appendix A) and the information obtained is converted into RDF using the USER ontology and Jena API, as well as being stored on the user's computer as an RDF file. Adaptation module is responsible from converting user's data into RDF and an example user profile is represented in Figure 6-8. Since the description of the user's interests and background is represented using semantic standards, the information about the user can be easily interconnected to the portal's contents.

6.9.2 Semantic Bookmarks

To avoid long questionnaires, only a few questions are asked during the registration process. Also, the added functionality of semantic bookmarking is available in SEMPort. Users can explicitly add semantic bookmarks to the ACM CCS topics they are interested in during the semantic navigation or SEMPort can implicitly add semantic bookmarks to the user profile.

6.9.2.1 Adding Semantic Bookmarks Explicitly

SEMPort recognizes ACM CCS topics throughout the semantic navigation, and if the user is logged on to the portal, ACM CCS topics are presented with bookmarks as shown in Figures 6-2 and 6-18. If the user is interested in the topic presented, she can simply click on the "add" button. The bookmark is then added to the user profile and a "successfully added" message is displayed.

6.9.2.2 Adding Semantic Bookmarks Implicitly by SEMPort

Semantic bookmarks can also be added to the user profile implicitly. SEMPort supports the tracking of the user's browsing, and if the user spends a certain amount of time on an ACM CCS topic, that topic is then automatically added to the user's profile. This functionality is supported by a Javascript function. If an implicitly added bookmark is incorrect, the users can control their profile from a Web front-end (Figure 6-21), which is provided for users to manage their profiles by adding, deleting or changing information.

6.9.2.3 Related Work

Semantic bookmarks are also used in other applications, such as SEAL (Maedche et al. 2002) and Magpie (Domingue et al., 2004). In the SEAL approach, semantic bookmarks are used for querying. Some queries can be saved as bookmarks and it is possible to use them later. In Magpie, a semantic bookmarking engine is used for serving related URIs. This engine assumes queries using domain-specific filters and presents related pages. In SEMPort, we use semantic bookmarks for the purpose of personalization; for the recommendation of related links.

6.9.3 A Web Front-end for Editing User Profiles

A Web front-end is provided for users to control their profiles. OntoWebber (Jin et al., 2001) and OntoWeaver (Lei et al., 2004; Lei et al., 2004b) also employ user models for customization. In these approaches, the user profiles are completely controlled by a privileged user. However, in SEMPort, users have control of their profiles. By using this Web front-end, users can change or delete information, and change the weights of their interests depending on the importance to them, such as low, medium or high interest. For example, Figure 21(a) shows an undergraduate student profile and Figure 6-21(b) that of a teacher profile, which represent Web interfaces for the editing of profiles.

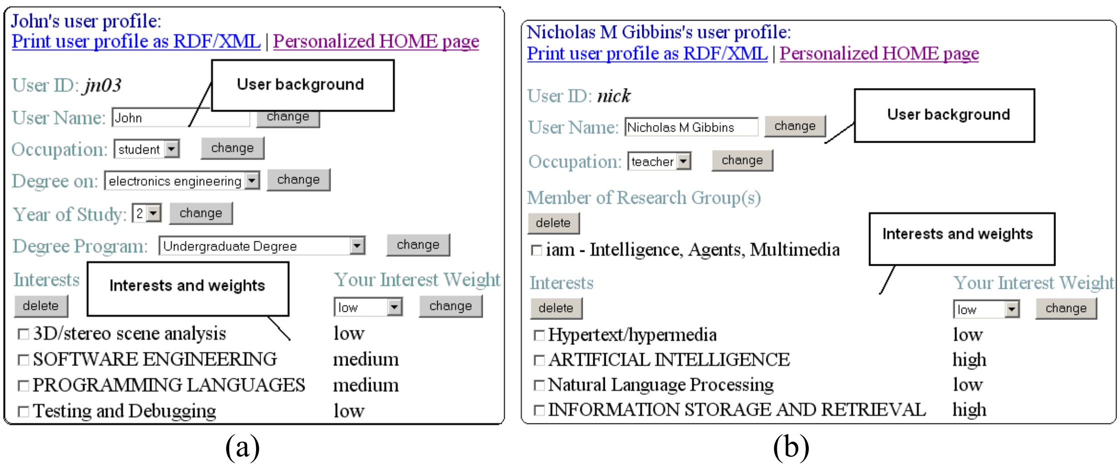


Figure 6-21 (a) A Web front-end for editing a student profile. (b) A Web front-end for editing a teacher profile

Users can easily access their profiles from the link provided at the upper frame, as shown in Figure 6-1. Note that all the changes to the user profile are maintained by the adaptation module. For example, when the user makes a request to change information in the profile, changes are first done by the adaptation module in a temporary user model using the Jena Ontology API and then the user profile is converted into RDF and stored in the user's computer as an RDF file. Following this, the information displayed in the user profile Web front-end is updated. In the same way, SEMPort can add implicit semantic bookmarks to the user profile by using the adaptation module. The adaptation pipeline is illustrated in Figure 6-22. Users can see the effects of the changes immediately from their user profile editing Web interface.

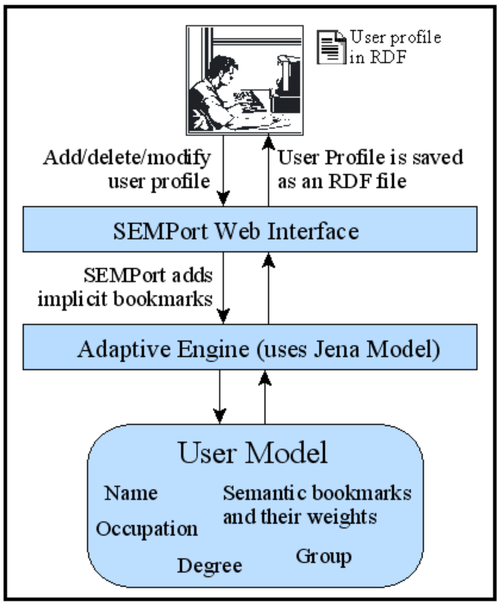


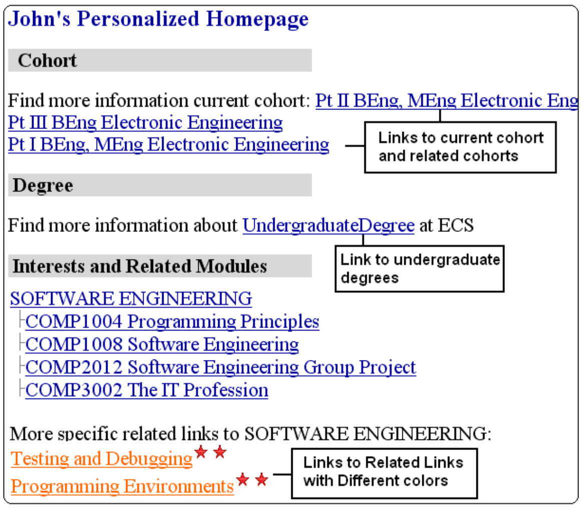
Figure 6-22 Interaction of the user with SEMPort for managing the user profile

6.10 Personalization Based on the User Model

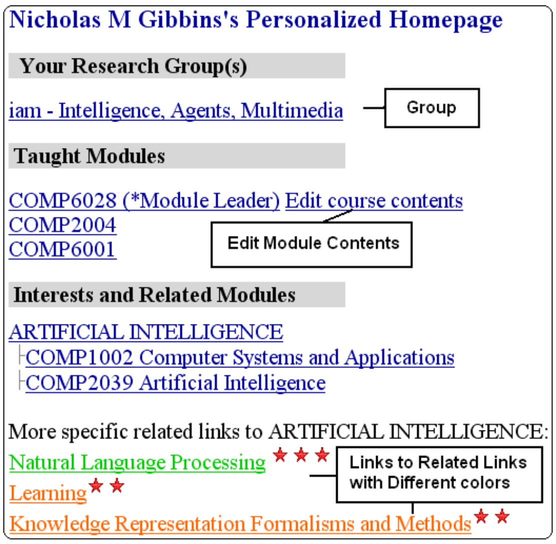
Generally portals contain huge amounts of information and users often have the feeling of being "lost in hyperspace." Therefore, users need to be guided to relevant information sources during the navigation. Most semantic portals do not provide any adaptation to the end-users, for example, the KAON portal (Ehrig et al., 2002), the OntoWeb portal (Spyns et al., 2002), ODESeW (Corcho et al., 2003; Corcho et al., 2006), MuseumFinland (Hyvonen et al., 2004; Hyvonen et al., 2004b). The Rewerse portal (Abel and Henze, 2005; Brunkhorst and Henze, 2005), SEAL (Maedche, 2001), and REASE (REASE, 2008) provide limited adaptation to the end-users. In SEMPort, our purpose is to support user navigation with personalized services; AH techniques can be supported by this design using ontology-based user models. In this section, we explain the provided personalization features.

6.10.1 Personalized Homepages

Personalized homepages are generated to guide users to relevant pages depending on their background and interests. When a user is logged in to SEMPort, a personalized homepage is automatically generated. The relationships between the user metadata and the domain model concepts are used to provide personalized views. The user can easily access the homepage from the link provided at the upper frame (see Figure 6-1).



(a)



(b)

Figure 6-23 (a) Personalized student homepage. (b) Personalized teacher homepage

In Figure 6-23, we illustrate examples of student (a) and teacher (b) homepages. Based on different user types, different contents and hyperlinks are shown in personalized homepages. For example, Figure 6-23 (a) is a student homepage; we provide links to the current cohort, related cohorts and undergraduate degrees, interested topics, modules that cover the interested topics and related topics. Also Figure 6-23 (b) is a teacher homepage; we provide links to the group, taught modules, interested topics, modules that cover the interested topics and related links. The taught modules are not explicitly defined by the teachers during the registration, but they are inherited: if the teacher is a member of ECS, the ECS ID of the teacher is used to inherit, or if the teacher is not a member of ECS, the full name of the teacher is used to inherit this knowledge. Also, in the personalized homepage, the interests of the users are ordered from highest rate (high) to lowest rate (low).

6.10.2 Annotation of Hyperlinks with Visual Cues

Related links in personalized homepages (Figure 6-23), and recommendation links (Figure 6-2) during the semantic navigation are annotated with different visual cues depending upon the similarity of a link to the user's interests. This is calculated using the equations below.

$$w_T(l) = \sum_{i=1}^{n} \left\{ \begin{array}{cc} w_i, & u_i = l \\ w_i \times 0.5, & \text{skos:broader}(l, u_i) \\ w_i \times 0.25, & \text{skos:broader}(u_i, l) \\ w_i \times 0.75, & \text{skos:related}(u_i, l) \\ 0, & \text{otherwise} \end{array} \right\} \tag{6.6}$$

$$s(l) = w_T(l) \Big/ \sum_{i=1}^{n} w_i \tag{6.7}$$

where $w_T(l)$ represents the total weight of link *l*, $s(l) \in [0,1]$ is the similarity of a link *l* to the user profile, *n* is the number of interests of a user *u*, and w_i is the associated weight of each interest. w_i can take values 1, 2 and 3, which indicates low, medium, and high interest respectively. The ACM CCS taxonomy is used for the calculation of the similarity (note that the deepest depth is three in this taxonomy). If the link matches with a user interest, then the weight of the interest is taken. If a user's interest is broader than the area of the link (i.e. the link is more specific than the user interest), then the weight is multiplied by 0.5. If the link is broader than the user interest (link is more general than the user interest), then the weight is multiplied by 0.25. If the link is a skos:related of a user interest, then the weight is multiplied by 0.75. If the link does not match any of the conditions explained above, then 0 is taken.

It should also be noted that the ratio values (0.25, 0.5 and 0.75) were selected by experimenting different value sets. With high ratio values, the similarity value was calculated high (>0.5) and with low ratio values, the similarity value was calculated low (<0.5). Best results were obtained with 0.25, 0.5 and 0.75; with these values, the similarity value could vary between 0 and 1. As a result of these, ratio values were selected as 0.25, 0.5 and 0.75.

Figure 6-24 shows how the weighting algorithm works. The user has an interest to portal:D.2 (Software Engineering). Based on the interest of the user and the relationships between different ACM CSS topics, the weights of the links are changing.

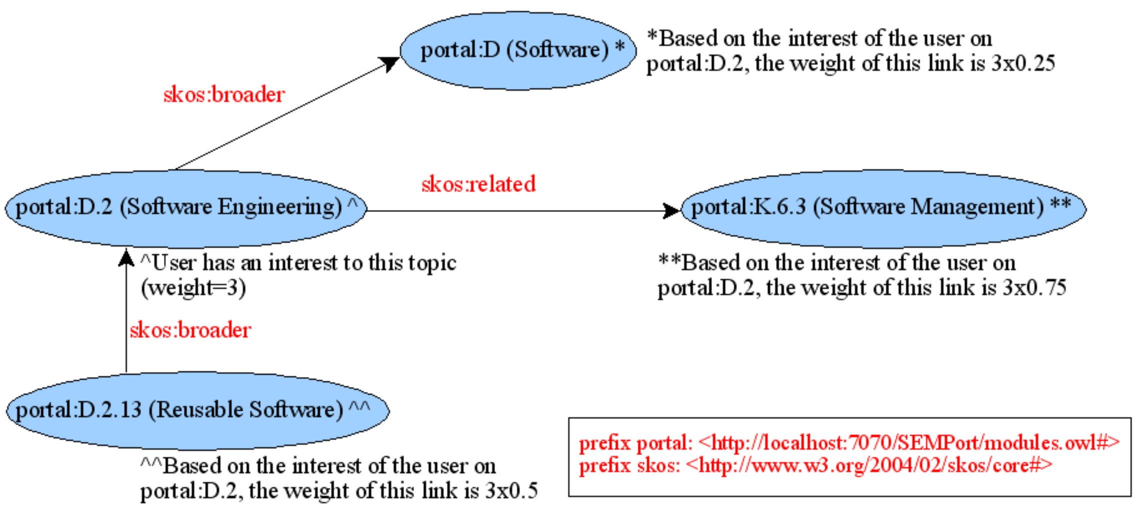


Figure 6-24 Illustration of weight calculation

Following this, similarity is thresholded as shown in Equation (6.8) and related links are presented with different colors. Green links represents *high similarity*, orange links represents *medium similarity*, yellow links represents *low similarity* and no color annotation is performed if the similarity is zero. For users with color deficits, the colored hyperlinks are presented with different number of stars (green – three stars, orange – two stars and yellow – one star). Related links are annotated with different visual cues even if the link does not contain the actual bookmark, and this is because of the use of semantics for describing semantic bookmarks. As a result of these, different users see different views.

$$
\text{link} = \begin{cases} \text{green}, & 1 \geq s(l) > 0.65 \\ \text{orange}, & 0.65 \geq s(l) > 0.35 \\ \text{yellow}, & 0.35 \geq s(l) > 0 \\ \text{normal}, & s(l) = 0 \end{cases} \tag{6.8}
$$

6.10.3 Link Sorting in the General View

Throughout the presentation of general views during semantic navigation, if the user is logged in to the portal, then the instances are sorted, based on their total normalized weights. If the instance is a module, then the ACM CCS topics covered by the module is used. For each ACM CCS topic covered by the module, a total weight is calculated by using Equation (6.7) based on the interests of the user. After the calculation of the weights for each ACM CCS topic, the total weight of the instance is calculated by summing all individual weights of ACM CCS topics that are covered by the module. Then, the total weight is normalized by dividing the total weight to the total number of weights of the user's interests using equation (6.9).

$$
w_{T\_nor}(inst) = \sum_{i=1}^{m} w_T(l_i) \Big/ \sum_{i=1}^{n} w_i \tag{6.9}
$$

Where, $w_{T\_nor}(inst) \in [0,1]$ is the total normalized weight of an instance, $w_T(l_i)$ is the total weight of a link l_i (ACM CCS topic covered by the instance), *m* is the number of ACM CCS topics covered by the instance, *n* is the number of interests of a user *u*, and w_i is the associated weight of each interest. Next, instances are re-ordered from highest to the lowest during the semantic navigation based on their total weight as shown in Figure 6-1, right.

6.11 Content Editing/Provision Web Interface

The process of content provision in semantic portals is difficult, especially where content is continuously changing. For instance, newly added information cannot be seen without restarting the server, as demonstrated in the KAON portal (Ehrig et al., 2002), and Web interfaces are difficult for the editing and the insertion of information, as demonstrated in ODESeW (Corcho et al., 2003). To address these problems, we developed an easy–to-use Web interface for ECS CMWP, for changing and updating instances in real-time. This Web interface can be used by module leaders and the administrator. Administrator and module leaders have different access rights; therefore, content provision and editing is covered in two sections.

6.11.1 Content Editing/Provision by Module Leaders

To allow distributed maintenance of the portal, we gave access rights to the module leaders to update the contents of the modules. Module leaders can access module contents from their personalized homepages. In personalized homepages, if the user is a module leader, a link is provided to access the contents of the module as shown in Figure 6-23 (b). When the user follows the link, a distributed Web front-end is opened; all attributes (including super-classes) of the selected module are listed for update as shown in Figure 6-25 (a). Users can easily add, delete or change attributes from this interface.

6.11.1.1 Attribute Addition

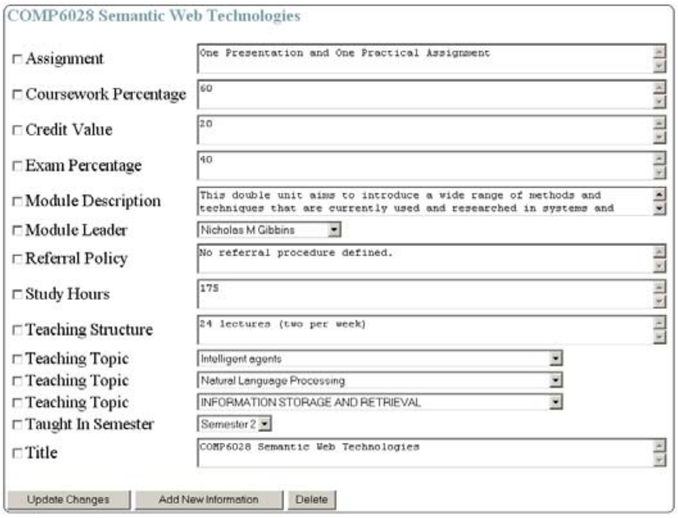
In order to add a new attribute to the current instance, the user has to press the "Add New Information" button (Figure 6-25 (a)). Then, all possible attributes (including the super-class attributes) are listed for selection as shown in Figure 6-25 (b). After selection of the attribute, the newly added attribute is shown on the editing interface, which allows the user to attach values. If the value of the newly added property is a literal, a text area is shown to fill in keywords, or if the value is a relationship, a drop-down list is displayed. All valid instance values (including those inherited) are added to the drop-down list (Figure 6-25 (c)), so the user does not have to know the details of the system. When the user selects the "Update Changes" button, the current values of the attributes are permanently saved to the KB. The changes can be seen without restarting the server and also a consistency check is performed and a report is presented (Figure 6-25 (d).

6.11.1.2 Attribute Editing

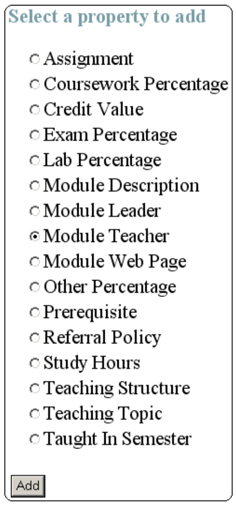
The interface also allows the editing of the attribute values easily. For the editing of relationship values, valid instance values are again added into a drop-down list as described above. If the user wants to change the value of an existing attribute, she can select a different value from a drop-down list for relationship properties and can update information from the text area for literal properties. Note that multiple values can be added to the same attribute. Changes are saved to the KB, when the “Update Changes” button is clicked. Changes can be seen in run-time without restarting the server and a consistency report is also provided.

6.11.1.3 Attribute Deletion

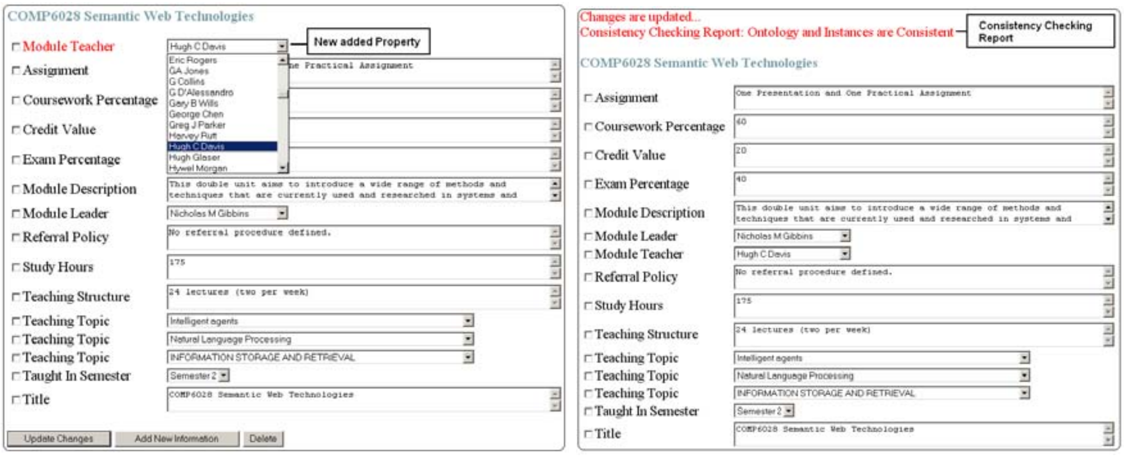
The interface also allows for the deletion of the attributes. Users can delete attributes and their values, simply selecting the checkboxes of the attributes and pressing the “Delete” button. Multiple attributes can be deleted at the same time. Again, changes are permanently saved to the KB. The changes can be viewed without restarting the server and a consistency report is also provided.



(a) SEMPort Web interface for content editing/provision: all attributes of ModuleInSession instance are listed



(b) SEMPort Web interface for content editing: addition of a new attribute



(c) SEMPort Web interface for content editing: addition of a new attribute value

(d) SEMPort Web interface for content editing/provision: update completed

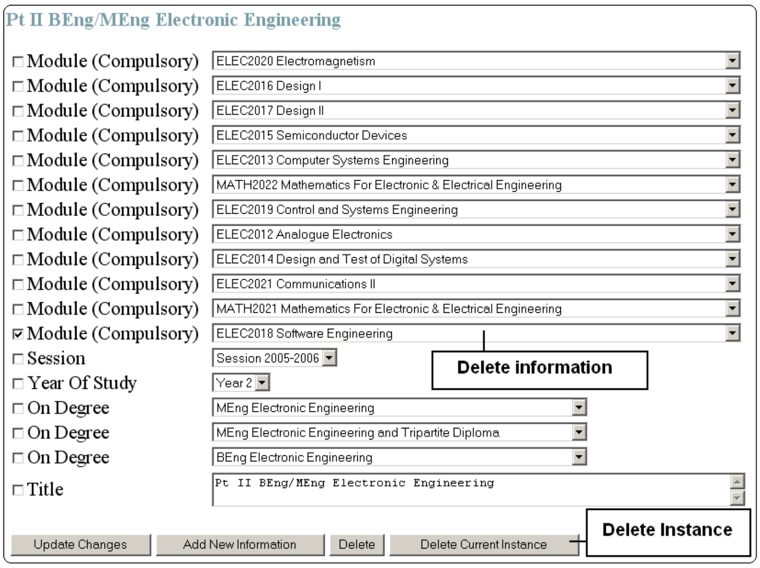
Figure 6-25 Illustration of content editing/provision interface

6.11.2 Content Editing/Provision by the Administrator

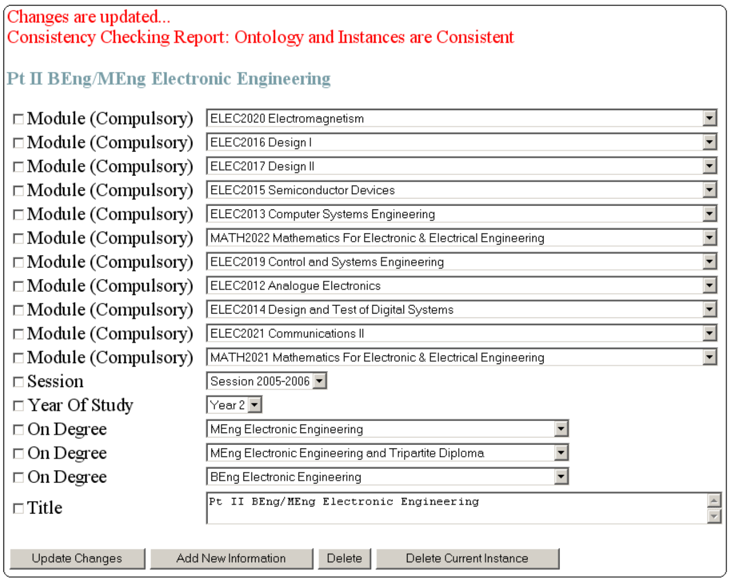
An administrator can edit *any* instance of the ontology using the content editing and provision Web interface. When the administrator is logged on to the portal, during the semantic navigation, for each instance of the ontology, a link is provided for the update of the contents (see Figure A-5 in the Appendix A). At the same time, the administrator can use semantic navigation and/or ontology-based search to find the instance. Once the instance is found, the administrator can update the contents of the instance from the link provided from the semantic navigation.

6.11.2.1 Attribute Addition, Editing and Deletion

The administrator can add new attributes, can update the value of existing attributes, and can delete attributes on *any instance* in the same way as explained above. In Figure 6-26 it can be seen that the administrator has marked for deletion of a compulsory module for the Pt II BEng/MEng Electronic Engineering cohort. Again all changes can be seen at run-time without regeneration, and a consistency report is displayed after changes are permanently saved to the KB.



(a) A Web interface for content editing/provision: one of the attributes of CohortInSession instance is selected for deletion



(b) A Web interface for content editing/provision: when deletion is completed

Figure 6-26 An example content editing/provision on a semantic instance

6.11.2.2 Instance Deletion

The administrator can also permanently delete an instance from the KB using the content editing and provision Web interface. An instance can be deleted by clicking the “Delete Current Instance” button. Before the removal of an instance from the KB, the KB is first interrogated for other instances that use the this instance. If the current instance is not used by other instances, then all of the attributes of the instance and the instance itself is deleted from the KB permanently. If the instance is used by another instance, the instance is not deleted and the cause of the action is explained together with the instances that use the current instance. In Figure 6-27, the administrator tried to delete the instance "COMP1004". However, because the instance is used by other instances, it could not be deleted. Instead, an explanation was given.

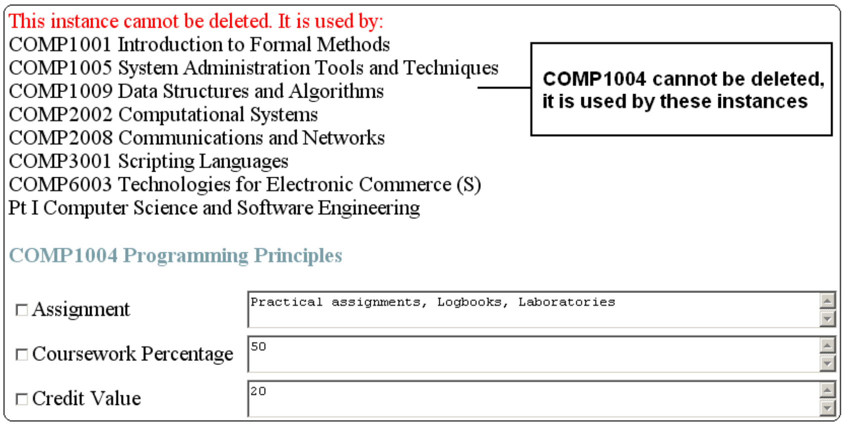


Figure 6-27 Deletion of an instance from the KB

6.11.3 Notes on the Content Editing/Provision

The content editing and provision Web interface can be used for the maintenance of different instances of different ontologies, since it is implemented independent of any ontology domain. As the contents of SEMPort can be changed easily, this interface can be used for different ontologies easily without adaptation. In addition, the content editing and provision Web interface can be used for maintenance by other ontology-based Web applications, since it can work on different browsers and can be accessed by other programs using Java. Also, we alleviate the problems of the editing of relationship values; valid instances are automatically inherited from the KB on the fly and presented to the user with descriptive titles. Therefore the user does not have to know the system-specific terms. Also, all the changes are saved permanently to the KB and can be seen at run-time without regeneration. For future work, instance addition will be provided from this interface.

6.12 SPARQL end-point

A Web front-end was created for the querying of the SEMPort's KB with SPARQL queries for users who are interested in SPARQL. We added the required namespace prefixes for the query area. On this page, a brief explanation of SPARQL is also given. Results are executed on the inferred knowledge and can be viewed in different formats, such as a numbered list and an HTML table, as shown in Figure 6-28. This interface allows users to flexibly query the contents of the SEMPort KB with SPARQL queries.

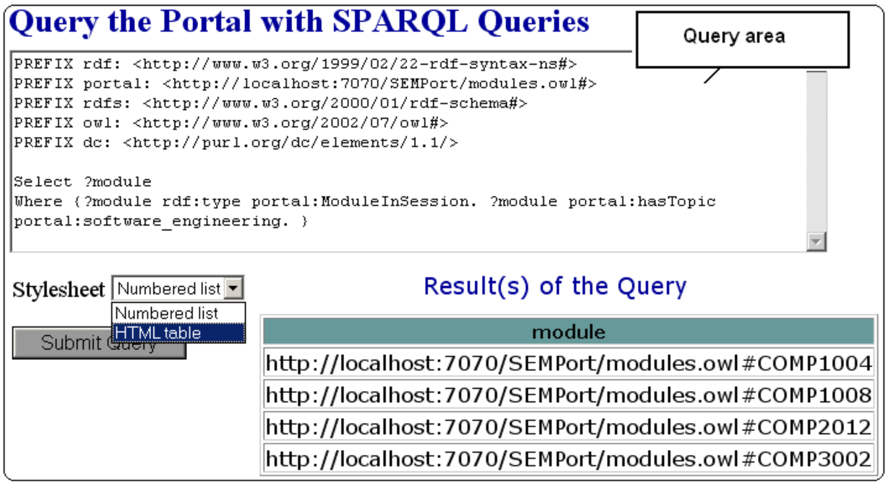


Figure 6-28 SPARQL End-point

6.13 Testing SEMPort on Other Ontology Domains

To allow the development of semantic portals with a low cost, SEMPort is implemented with re-usable components. Thus, different ontologies and instances can be used. To illustrate this, we have used W3C's Wine ontology[50]. The Web front-end is utilized to upload the ontology file to the KB (see Figure 6-29). Figures 6-30 and 6-31 show semantic navigation in general view and in detailed view respectively using the Wine Ontology. Also, users can perform ontology-based search using the properties of this ontology as shown in Figure 6-31. Since the personalization is specific to the ECS CMWP domain, the user ontology should be changed. Finally, Figures 6-32 and 6-33 illustrate the use of the content editing/provision interface on the Wine Ontology (Wine Ontology, 2003).

[50] www.w3.org/TR/owl-guide/wine.rdf [last accessed, 19/11/2008]

Figure 6-29 Web front-end loading the Wine Ontology to the KB

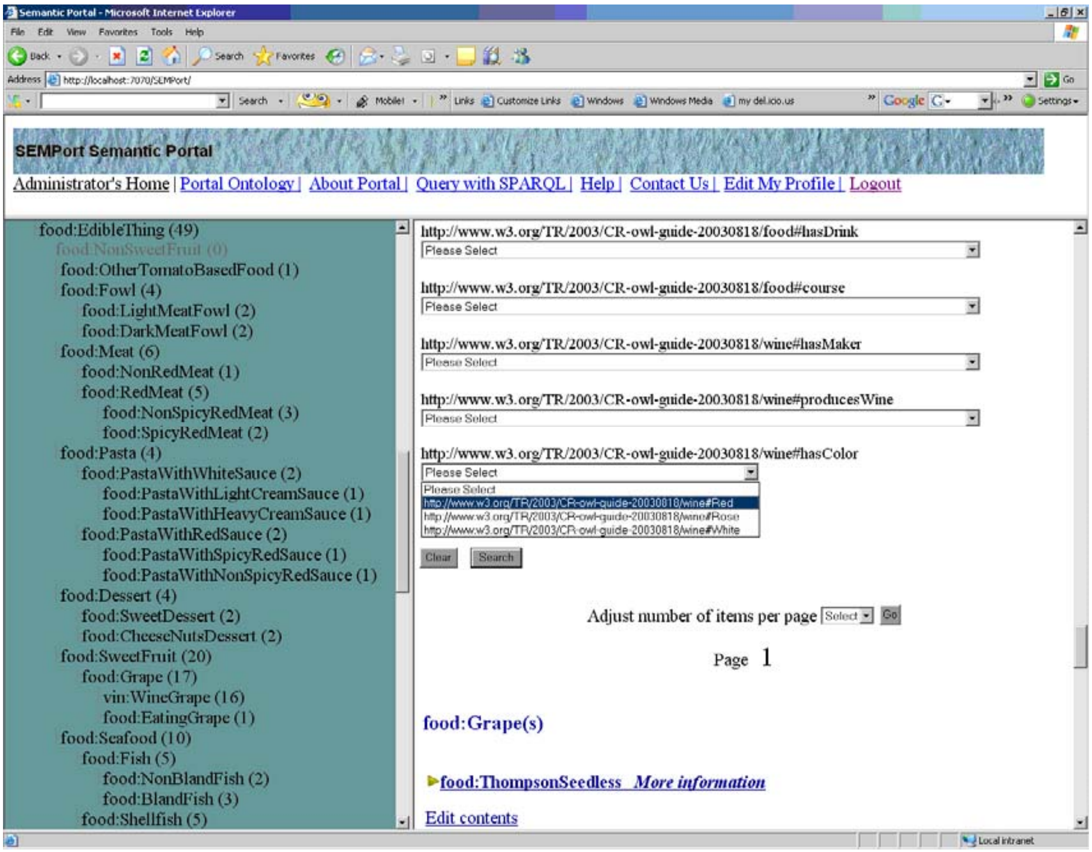


Figure 6-30 A general view from the semantic navigation using the wine ontology

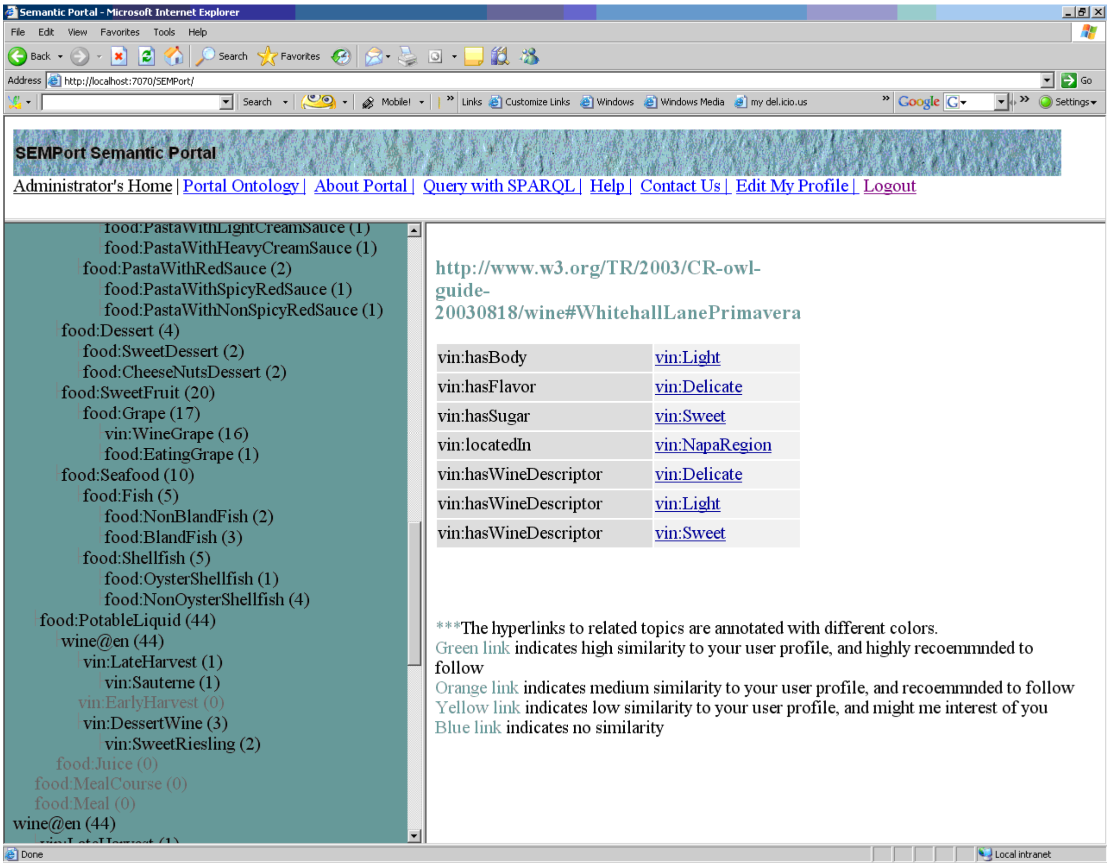
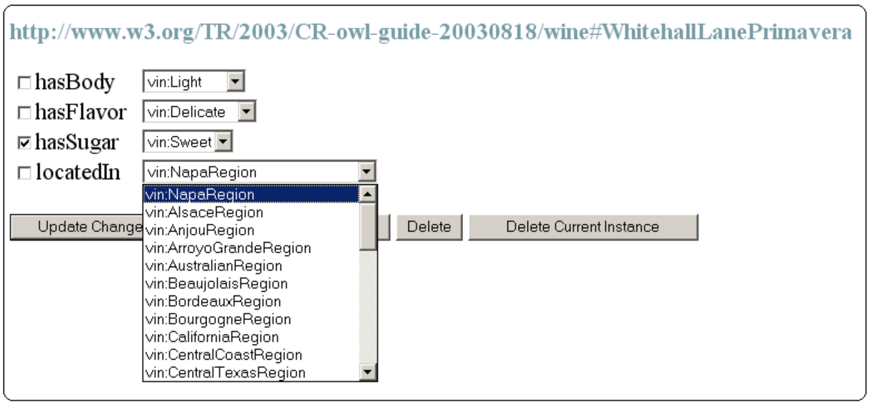
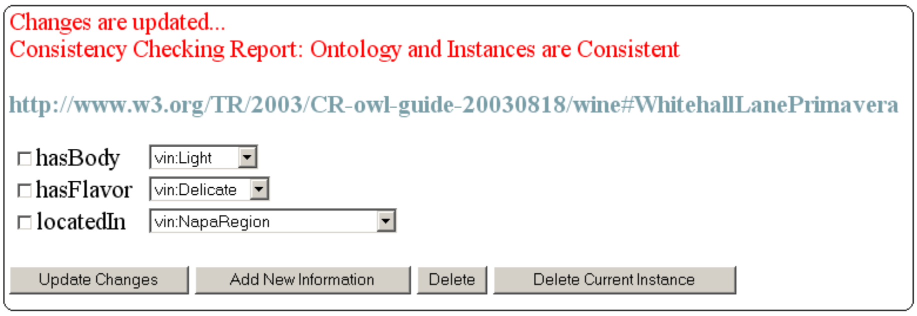


Figure 6-31 A detailed view from the semantic navigation using the wine ontology

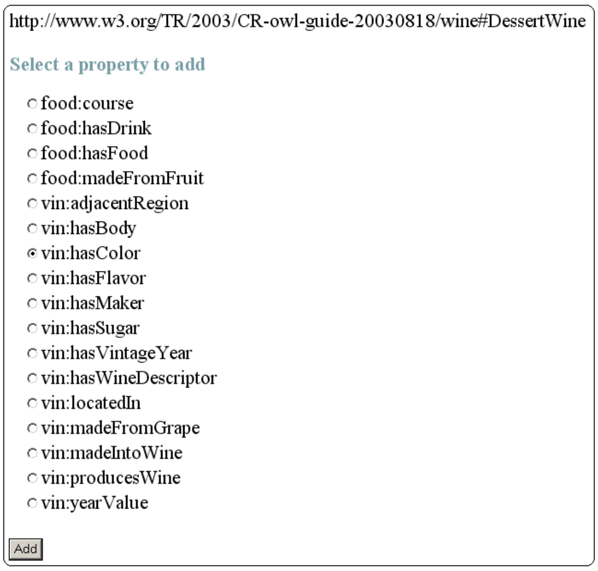


(a) Content editing and provision interface for the update and deletion of attributes

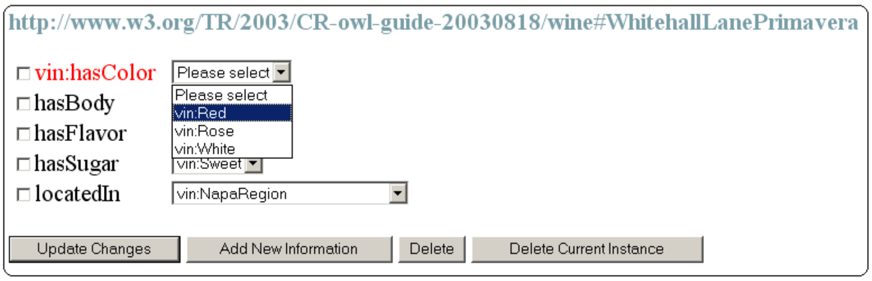


(b) Content editing and provision interface after the update and deletion of attributes

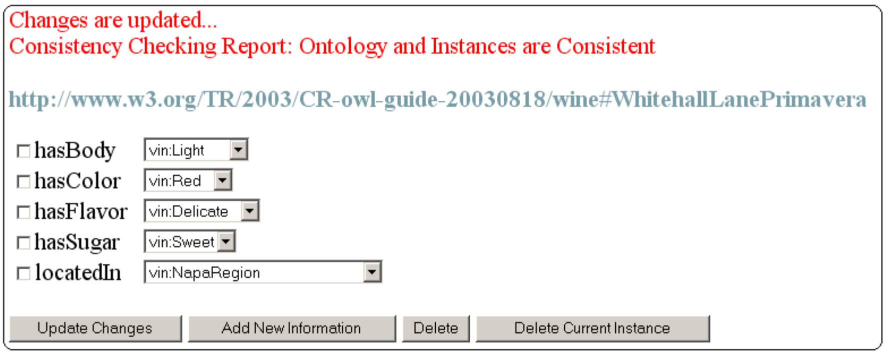
Figure 6-32 Illustration of attribute deletion and edition using the wine ontology



(a) A List of properties that can be added to Dessert instance



(b) The newly added property was appeared with valid available instances



(c) Content editing and provision interface after the addition of the attribute

Figure 6-33 Illustration of an attribute addition using the wine ontology

6.14 Chapter Summary

This chapter discussed the system design, the user modelling and the functionalities of the proposed semantic portal, SEMPort in detail. In our approach, we aimed to alleviate problems associated with content provision and provided three different mechanisms for the maintenance of the portal in real-time. Additionally, to improve the user's browsing, enhanced semantic hyperlinks are created, which we call explicit, inverse, implicit and recommendation links. Furthermore, to adapt the portal's contents to different users, we applied an ontology-based user modelling architecture and provided adaptive navigation and adaptive presentation. On the other hand, we proposed a generic architecture, so that it can be adapted to different domains at a low cost. Except for personalization (user model), all of the features of SEMPort are domain independent (ontology neutral) and can be easily adapted to different ontologies and domains as illustrated on the Wine Ontology. It is also noted that SEMPort is intended to be domain independent (ontology neutral) and it is not evaluated against multiple ontologies or interoperability of ontologies. In the next chapter, we discuss the evaluations that were undertaken using SEMPort.

7 Evaluation of SEMPort

The proposed semantic portal, SEMPort, was evaluated using structured review and user-based empirical study. In this chapter, we discuss the evaluations that were undertaken.

7.1 Structured Review

Evaluation is a part of the design-evaluate-design (an iterative design of a hypermedia system based on results of repeated evaluations) cycle of a hypermedia system. According to Nielsen and Molich "*the evaluation is concerned with gathering information about the usability or potential usability of a system in order to improve features within an interface and its supporting material or to assess a completed interface*" (Preece, 1993). Therefore, for improving and assessing a system, user evaluations need to be undertaken during the development cycle. In order to assess the usability of SEMPort a structured review was performed.

7.1.1 What is Structured Review?

Structured review is used to help define the interface problems of a system and used for improving user interface aspects. It is a part of the design-evaluate-design cycle of a hypermedia system. In a structured review, reviewers are asked to use a system and describe the potential problems that they foresee arising. The form of reporting adopted by reviewers can differ, for example structured reporting, unstructured reporting and predefined categorization can be used (Preece, 1993). All these styles have their advantages and disadvantages. Structured reporting is easy to analyze but inhibits spontaneous suggestions. Unstructured reporting is difficult to analyze but invites spontaneous comments and suggestions. In predefined categorization, reviewers are given a list of problem categories, and they report the occurrences of the problems; this is therefore very easy to analyze, but completely inhibits spontaneous comment and advice. In our research, structured reporting was adopted because it was easy to analyze. A predefined form was given to the reviewers, but additional comments were also encouraged to capture spontaneous advice.

Heuristic Evaluation: Structured review can be guided by general usability principles, known as heuristic evaluation (Neilsen and Molich, 1990). Heuristic evaluation, first developed by Nielsen and Molich, is a method of structuring the critique of a system using a set of relatively simple and general heuristics. The idea behind heuristic evaluation is that several evaluators are independently asked to comment on an interface design using a list of general usability principles. These usability principles are referred to as the "heuristics". The results of the individual evaluators are then aggregated to a list of potential usability problems of the design. The heuristics were first defined by Molich and Nielsen (Nielsen and Molich, 1990; Molich and Nielsen, 1990), and were later updated (Nielson, 1993; Nielson, 1994) based on ongoing research into heuristic evaluation.

Nielsen and Molich recommend that heuristic evaluation be done with between three and five evaluators, since using more does not have a significant effect on finding more usability problems (Neilsen and Molich, 1990). Usually five evaluators can identify 75% of the usability problems. Heuristic evaluation is also preferred because the study is relatively cheap and cost-effective (Jeffries et al., 1991). Hence it is often considered as a discount usability engineering method. The following list shows the heuristics typically used in a heuristic evaluation (see Figure A-6 in the Appendix A for the explanations).

- Visibility of system status
- Match between the system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalism of the design

- Help for users to recognize, diagnose, and recover from errors
- Help and documentation

In this research, some form of structured reviewing was used, in particular discount usability engineering. The method of discount usability engineering was first proposed by Nielsen and Molich (Molich and Nielsen, 1990; Nielsen, 1992). The procedure is based on the use of three techniques:

- Scenarios
- Simplified thinking aloud
- Heuristic evaluation

Thinking aloud is a usability engineering method that involves having a subject use the system while continuously thinking aloud (Nielsen, 1993). The observer is thus enabled to understand how the users view the computer system by verbalizing their thoughts. Discount usability engineering is a hybrid of empirical usability testing and heuristic evaluation. The features that result in important discount include: the scenarios are small and can be changed easily, the thinking aloud method is done informally and the whole cycle needs few reviewers, since the number of additional usability problems found by more reviewers was not worth the extra effort.

7.1.2 The Procedure Used

The evaluation was carried out on the initial version of SEMPort. To understand the usability problems of the portal, discount usability evaluation was used. Four evaluators (three PhD students and a member of research staff who has a background in human computer interaction) were used in the evaluation and none of the reviewers had seen or used SEMPort before. During the evaluation, scenarios were used to test the interface, thinking aloud was done informally, and reviewers were also asked to comment on the ten heuristics of Nielsen (Nielsen, 1993). We also asked reviewers to comment on the positive aspects of the portal, and these comments were recorded by the observer. User instructions and heuristic evaluation commentaries are attachted to the Appendix B.

The scenarios were chosen to ensure that the reviewers would visit the different user interfaces of the system (i.e. semantic navigation, ontology-based search, personalization, and content editing and provision). The reviewers were asked to look for and comment on the flow of information from screen to screen and on each screen to evaluate the usability. While the reporting system used can be classified as 'structured reporting', a predefined form (See Figure A-7 in the Appendix A) was used to record the usability problems found using Nielsen's heuristics. The evaluation was performed one reviewer at a time, and a thinking aloud approach was encouraged to allow observer to note the comments of the evaluators.

7.1.3 The Results

The Usability Problems Found by the Evaluators: We used the ten essential heuristics suggested by Nielsen for finding the usability problems of the design. In the remainder of this section, we summarize the usability problems found, which are grouped under Nielsen's ten principles. Nielsen's principles are shown in bold, while the bullet points present the problem number and the general principle distilled from the evaluation. Note that evaluators did not find any usability problems for the usability heuristic "aesthetic and minimalist design". They found that all the information presented was relevant and useful.

Visibility of system status

- **1:** "+" and "–" buttons on the left frame are not visible. It is not seen that the concepts can be widened or narrowed using the buttons. Buttons should be presented larger.
- **2:** At the ontology hierarchy, the meanings of numbers in brackets are not clear.
- **3:** During the detailed view of an instance, all attributes should be presented in alphabetical order. Otherwise users may be confused where to look at.
- **4:** After addition of a bookmark, it would be better to inform the user that the bookmark was added successfully.
- **5:** After the completion of registration at personalization, it would be better to inform the user that the registration was completed successfully.
- **6:** Searches are sometimes slow, and the user could be informed that the search may take some time.

- **7:** Users can edit their profiles from the link provided at the upper frame. For instance, the link is labeled as “John’s Profile”. But since it is not clear that users can edit their own profiles, the link’s name can be changed to “Edit My Profile”.
- **8:** In the content editing and provision interface, for the editing of literals, text boxes are used. However, when there are more characters in the text box, the information cannot be seen and it is difficult to edit. Instead of text boxes, text areas could be used.

Match between the system and the real world

- **9:** Use only descriptive titles for ontology concepts in the ontology hierarchy, do not use technical terms or short URIs that have no relevance to the user.
- **10:** Use only descriptive titles for attributes in the detailed view, do not use technical terms or short URIs that have no relevance to the user.
- **11:** During the detailed views of a module, the meaning of the “Prerequisite (other)” is not clear. Its meaning should be explained clearly; prerequisite of prerequisite can be used for explanation.
- **12:** In the content editing and provision interface, for the addition of new information the “Add New Property” button is used. However, non-computer science users will not understand the meaning of the button and its title should be changed to “Add New Information”.

User Control and Freedom

- **13:** There is no way to exit from the personalized homepage. A list of links could be provided for users to go other functionalities of the portal.
- **14:** In the personalized homepage, the layout is a little bit confusing. Allow users to select what to put on the homepage.

Consistency and standards

- **15:** The names of the ACM CCS topics are sometimes presented with capital letters and sometimes with lower case letters; it should be consistent.
- **16:** Colors of the personalized hyperlinks are very close to each other. For example, there is little difference between orange and yellow. More distinguishable colors could be selected.

Error prevention

- **17:** During navigation, "null error" appeared and the presentation disappeared; this problem should be solved.
- **18:** During navigation, an implicit bookmark was added even when the user was logged out. It should only add bookmarks if the user is logged in.

Recognition rather than recall

- **19:** The meanings of the different visual cues on the hyperlinks for the personalization are not clear. Their meanings should be explained in the page presented.
- **20:** Users may lose which concept they are inside during the navigation, thus it is better to highlight the name of the concept in the ontology hierarchy, where the user is currently located.

Flexibility and efficiency of use

- **21:** Opening of a link from an ontology hierarchy may take long (e.g. 5 seconds), and impatient users may try opening the link many times. Provide a message that the link is opening.
- **22:** The users may want to add bookmarks from a list of topics. Provide a New Topic Addition button from their profile editing Web interface.

Help users recognize, diagnose and recover from errors

- **23:** In the search, when the result of the search is empty, there should be a "No result found" message, instead of no explanation.

Help and documentation

- **24:** There is no in-context help or keyword search for help topics.

In summary, a number of observations were made during the evaluation, the majority of which dealt with improvements to the user interface.

Re-Designing After the Evaluation: Before the empirical user study, some of the usability problems found by the evaluators were fixed. Problems 1, 4, 5, 7-10, 12, 17-19 and 23 were completely corrected. Problem 3 was partially solved; direct properties of the instance were ordered alphabetically, however, indirect (inferred) properties could not be ordered. Because of the limited time before the empirical study, the remaining usability problems could not be fixed. However, most of the problems related to the "visibility of the system status" and "match between the system and the real world" were fixed and this improved the look and feel of the interface.

7.1.4 The Comments on the Procedure

The evaluation was performed by one evaluator at a time. The reviewers took between one and one and half hours to complete the evaluation. During this time, the reviewers' comments were also noted.

7.2 Empirical Study

A two-phased experiment was set-up based on ECS CMWP and SEMPort, to measure users' performances for a set of tasks, and questionnaires were used to understand the users' attitudes to both systems. In the first part, we study the implications of semantics in SEMPort, where subjects were asked to perform a set of tasks using ECS CMWP and a set of tasks using SEMPort. In the second part, subjects were asked to perform a set of tasks using SEMPort. During the experiment, users were asked to write down correct answers to all tasks. A thinking aloud approach was used to understand the user's way of thinking when performing a particular task and these observations were noted by the evaluator (users were encouraged by the evaluator to think aloud). Finally, in order to understand user attitudes to both systems, a post-questionnaire was used. Evaluation tasks and the original questionnaires are attached to the Appendix B.

In the experiment, ten participants took part. Participants were PhD students of different ECS research groups (i.e. LSL: 3, ISIS: 3, IAM: 4) who had varied individual research directions and different computer skills. In addition, two of the participants did ECS undergraduate degree and they were familiar with the ECS CMWP. Three of the participants were demonstrators and they also had experience with the ECS CMWP.

7.2.1 *Phase 1 – An Experiment on ECS CMWP and SEMPort*

To compare the difference between the semantic and non-semantic systems, we conducted an experiment on ECS CMWP and SEMPort. In this phase, our aim was to compare the two systems in terms of navigation. It should be noted that SEMPort uses the same information as ECS CMWP, but in SEMPort this data is represented with ontologies.

7.2.1.1 The Procedure Used

In the experiment, the ten participants were asked to carry out three different tasks using ECS CMWP and SEMPort. To remove the learning effect, the ten users were randomly divided into two groups. Group A users performed the three tasks on ECS CMWP first and Group B users performed the three similar tasks on SEMPort first. The task sets are listed in Table 7-1. The groups were then swapped round and the same tasks were repeated on the other system. During the experiment, the thinking aloud method was used and users were asked to write down the correct answers. We measured the task completion times for the different groups (no time limit was placed). In addition, a questionnaire was used after the experiment to identify users' thoughts on ECS CMWP and SEMPort in terms of navigation.

Table 7-1: List of tasks that were used in phase 1

SET A	SET B
1. Find the module leader and module teacher(s) of the course COMP2004	1. Find the module leader and module teacher(s) of the course COMP1007
2. List the course codes of compulsory modules for "I BEng Electronic Engineering" cohort	2. List the course codes of compulsory modules for "II BEng Electronic Engineering" cohort
3. Find the courses which have prerequisite COMP1003	3. Find the courses which have prerequisite COMP1004

7.2.1.2 The Results of Phase 1

The Results of Correct Answers: For each subject, the total number of correct answers in each group of tasks was calculated and a score generated. Note that each task contains a number of correct answers, and during the scoring each individual correct answer was counted. The results are illustrated in Figure 7-1. They show that participants performed better using SEMPort (98% satisfaction) compared to ECS CMWP (41% satisfaction). Eight out of the ten subjects answered all questions correctly (22) using SEMPort. On the other hand, subjects found fewer answers using ECS CMWP. The main reason for this was the structure of the information on ECS CMWP. For instance, in Set A, question 2, users were asked to find compulsory modules for the first year Electronic Engineering cohort. This information was available on the syllabus page of each course and users could navigate to syllabus pages from a list of alphabetically ordered courses. In order to find a cohort name, participants had to open all course modules one by one. Therefore, most of the participants either quit the question or used their background knowledge (i.e. students had an electronics degree assumed possible course modules) or logic. In the same way, in Set A, question 3, participants had to check each module for prerequisites. Again, most of the participants either quit or used their background knowledge. In contrast, in SEMPort, users could reach the different cohorts from the ontology hierarchy and could find prerequisite modules by simply using inverse links.

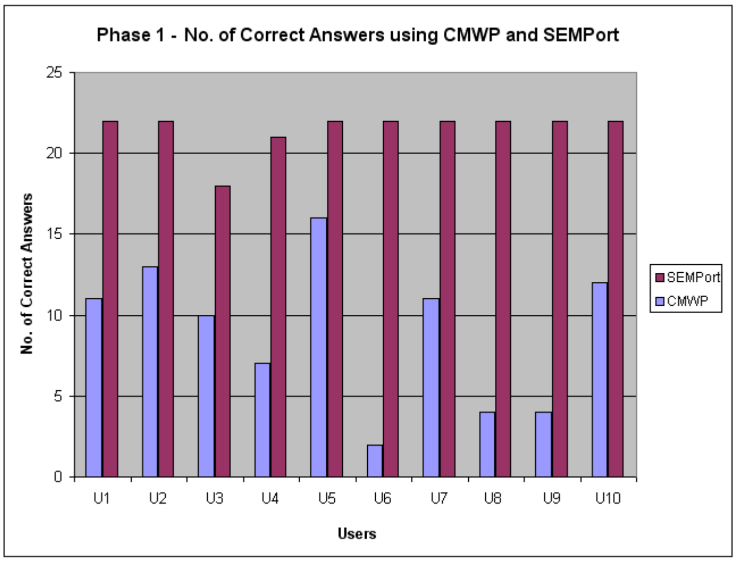


Figure 7-1 The total number of correct answers of the participants using ECS CMWP and SEMPort

Based on these observations, most users found questions 2 and 3 in Set A very difficult, because, the navigational structure of ECS CMWP does not allow users to browse it effectively. However, in SEMPort, the ontology hierarchy allows a good structure for the presentation of the contents. Participants also liked the extensive links (explicit links, inverse links, implicit links, and recommendation links) between different information items, because it enabled them to complete tasks easily.

The Results of Task Completion Times: Task completion times both on ECS CMWP and SEMPort is shown in Figure 7-2. Because most of the subjects quit question 2 or 3 in set A, task completion times for ECS CMWP were faster, despite the fact that the numbers of correct answers found using ECS CMWP was lower compared to SEMPort. Note that in both cases, all participants carried out their tasks on the same machine.

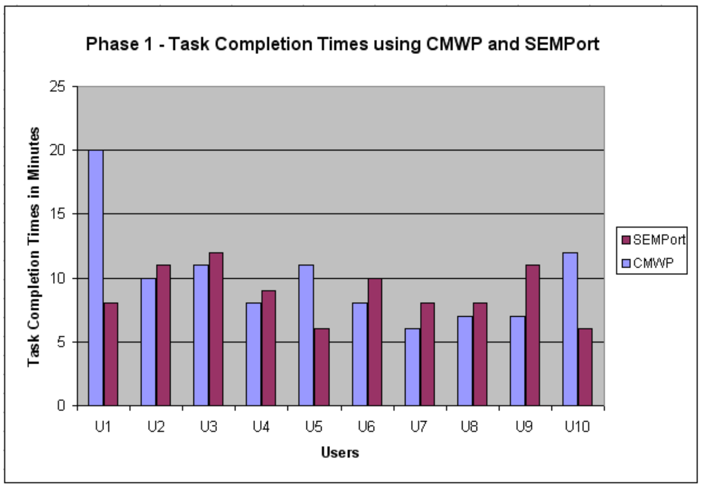


Figure 7-2 The task completion times on ECS CMWP and SEMPort

The Results of the Questionnaire for Navigation: To understand users' attitudes to navigation in ECS CMWP and SEMPort, a questionnaire was used. In the questionnaire, Likert-scale questions were used: a 5 point-scale ranged from the lowest 1 to highest 5, with comments associated with each answer. The results of the questionnaire are listed in Table 7-2 and show that participants rated "*How well were you able to complete tasks*" 1.7 for ECS CMWP and 4.5 for SEMPort. The correct answers of the subjects also showed this. Users found presented hyperlinks useful (4.5) and they believed that their navigation was improved with SEMPort (4.4) compared to 2.5 with ECS CMWP. Overall, subjects were satisfied with the semantic navigation in SEMPort with an 4.4 on average, compared to 1.9 of ECS CMWP. However, the speed of the semantic navigation was not found to be fast enough. The main reason was the use of the reasoning and this problem will be alleviated in the future.

Table 7-2: Post-questionnaire results for navigation of ECS CMWP and SEMPort

	ECS CMWP (Mean)	**SEMPort (Mean)**
I am easily able to use navigation?	2.9	4.2
How difficult was it to find information using navigation?	1.9	4.4
How well were you able to complete tasks using navigation?	1.7	4.5
How was the speed of the navigation?	3.2	3.5
How useful did you find presented hyperlinks?	2.7	4.5
Navigation improved my browsing facilities	2.5	4.4
Overall, how well were you satisfied with the navigation?	1.9	4.4

7.2.2 Phase 2 – An Experiment on SEMPort

To test the functionalities of SEMPort and the user's attitudes to the different aspects, we conducted an experiment on ontology-based search, personalization and content editing/provision. In this experiment, the same ten subjects were used and they performed six more tasks using SEMPort's interface. These tasks are listed in Table 7-3. Tasks one, two and three were used to calculate the score. Tasks four, five and six do not have specific answers: they were used by the subjects to analyze the interface of SEMPort, and therefore these questions were not used in the scoring. No time limit was placed on answering the questions. During the experiment, the thinking aloud method was used and users were asked to write down the correct answers.

At the end of the tasks, to understand individual views of the participants to different functionalities of SEMPort, questionnaires were used. We prepared different questionnaires for ontology-based search, personalization and content editing/provision. Likert-scale questions were used: a 5 point-scale ranged from the lowest 1 to highest 5, with comments associated with each answer.

Table 7-3: List of tasks that are used in the phase 2

1. Find courses that teach “artificial intelligence” and are taught in “Semester 2”
2. Find a course called “speech processing”, and list all the prerequisites of it
3. Find the course COMP1008 and then list other courses which might be relevant to this course (i.e. modules that teach similar or the same topics)
4. Do two tasks of your own
5. To analyze the personalization a set of tasks were used, i.e. registration, editing of profile, bookmark addition and inspection of homepages and navigation.
6. To analyze the content editing interface, a set of tasks were used, i.e. addition, deletion and update of information.

7.2.2.1 The Results of Phase 2

The Results of Correct Answers: For each subject, the total number of correct answers to the first three tasks was calculated and a score generated. Note that each task contains a number of correct answers, and during the scoring each individual correct answer was counted. The results are illustrated in Figure 7-3.

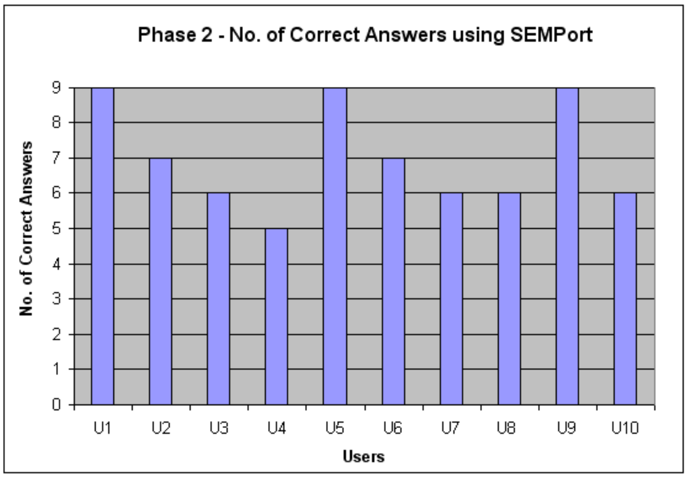


Figure 7-3 The total number of correct answers of the participants using SEMPort in Phase 2

The results showed that three out of ten subjects answered all questions correctly (9 correct answers). Task 3 was found difficult to understand by the subjects, and most of the participants only answered this question partially. In this question, users were supposed to find module COMP1008 first, then they could find similar courses by looking at the teaching topic of the module or related topics from the recommendation links. By using inverse links, they could see the modules which teach the same topics as COMP1008 or are teaching similar topics.

Questionnaire for the Ontology-based Search: The results of the ontology-based search questionnaire are shown in Table 7-4. The results showed that the searches were found easy to use with rating 4.3. Individual views also showed that the searches were not fast enough with 3.6. Overall, subjects were satisfied with ontology-based search with rating 4.4 out of a possible 5 and they preferred to have ontology-based search on ECS CMWP. Observations also show that participants enjoyed using concept-based searches; most of the subjects verbally indicated their interest in ontology-based search during the experiment.

Table 7-4: Questionnaire results for the ontology-based search

	1	**2**	**3**	**4**	**5**	**(Mean)**
How difficult was it to find information using the search?	- *(very difficult)*	- *(difficult)*	- *(unsure)*	7 *(easy)*	3 *(very easy)*	4.3
How was the speed of the searches?	- *(very slow)*	2 *(slow)*	1 *(fine)*	6 *(fast)*	1 *(very fast)*	3.6
Overall, how well were you satisfied with the search and do you want this kind of search at ECS CMWP?	- *(definitly no)*	- *(no)*	1 *(don't know)*	4 *(yes)*	5 *(definitly yes)*	4.4

Questionnaire for Personalization: The results of the questionnaire are shown in Table 7-5. The results show that the length of the registration was found to be fair (3.2). One of the aims of the use of semantic bookmarks is to reduce registration time. Users can extend their profiles using semantic bookmarks later. The participants also found the editing of the user profile easy (4.4) as was the addition of semantic bookmarks (4.2). Subjects rated 4.4 the usefulness of the personalized homepages, which shows that they liked the information presented. Although users rated reordering of the contents 4.0, and the hyperlinks with visual cues 4.1, these ratios are smaller compared to the homepage mean. However, two users rated reordering very useful (5) and four users rated hyperlinks with visual cues very useful (5). We can conclude that different users have different likes and in general all personalization features rated greater than 4.0. Overall, subjects were satisfied with the personalization with an average rating of 4.5 and preferred to have personalization on ECS CMWP. Observations also show that participants enjoyed personalization. Most of the subjects liked personalized homepages and verbally indicated their interest during the experiment.

Table 7-5: Questionnaire results for the personalization

	1	**2**	**3**	**4**	**5**	**(Mean)**
The length of the registration was short	- *(too long)*	1 *(long)*	6 *(fair)*	3 *(short)*	- *(too short)*	3.2
How easily were you able to edit your profile?	- *(very difficult)*	- *(with some difficulty)*	- *(unsure)*	6 *(easy)*	4 *(very easy)*	4.4
How easily were you able to add bookmarks?	- *(very difficult)*	- *(with some difficulty)*	- *(unsure)*	8 *(easy)*	2 *(very easy)*	4.2
How useful did you find information and hyperlinks on personalized homepage?	- *(not at all)*	- *(not too useful)*	- *(don't know)*	6 *(useful)*	4 *(very useful)*	4.4
How useful did you find reordering of contents during navigation?	- *(not at all)*	1 *(not too useful)*	- *(don't know)*	7 *(useful)*	2 *(very useful)*	4.0
How useful did you find the hyperlinks with visual cues?	- *(not at all)*	1 *(not too useful)*	1 *(don't know)*	4 *(useful)*	4 *(very useful)*	4.1
Overall, how well were you satisfied with the personalization and do you want this kind of personalization at ECS CMWP?	- *(definitly no)*	- *(no)*	- *(don't know)*	5 *(yes)*	5 *(definitly yes)*	4.5

Questionnaire for the Content Editing/Provision: The results of the questionnaire are shown in Table 7-6. The results showed that by using the content editing/provision Web interface, subjects were easily able to update, add and delete information and rated these features 4.5, 4.5 and 4.4 respectively. They also found the speed of updating was fast (4.0). Overall, subjects were satisfied with the content editing/provision Web interface with a mean rating of 4.4. Based on these observations, we also want to comment on the real-time operability of the interface. Subjects were really impressed that they could see the effects of the changes, by simply updating the information.

Table 7-6: Questionnaire results for the content editing/provision Web interface

	1	**2**	**3**	**4**	**5**	**(Mean)**
How difficult was it to change information?	- *(very difficult)*	- *(difficult)*	- *(unsure)*	5 *(easy)*	5 *(very easy)*	4.5
How difficult was it to add new information?	- *(very difficult)*	- *(difficult)*	- *(unsure)*	5 *(easy)*	5 *(very easy)*	4.5
How difficult was it to delete information?	- *(very difficult)*	- *(difficult)*	- *(unsure)*	6 *(easy)*	4 *(very easy)*	4.4
How was the speed of the update?	- *(very slow)*	- *(slow)*	- *(fine)*	10 *(fast)*	- *(very fast)*	4.0
Overall, how well were you satisfied with the content editing interface?	- *(very dissatisfied)*	- *(dissatisfied)*	- *(undecided)*	6 *(satisfied)*	4 *(very satisfied)*	4.4

Questionnaire for the Features of SEMPort: In the questionnaire, we also asked subjects "*Which feature of SEMPort did they like most?*" One subject preferred all the functionalities of the portal. The answers of the participants are represented as a pie chart in Figure 7-4. 8% of the subjects liked the content editing/provision Web interface, 23% of the subjects preferred ontology-based search, 31% of the subjects liked links (explicit, inverse, implicit and recommendation links) and 38% of the subjects preferred personalization. This result showed that there is more interest in the personalization and semantic hyperlinks compared to other functionalities of SEMPort.

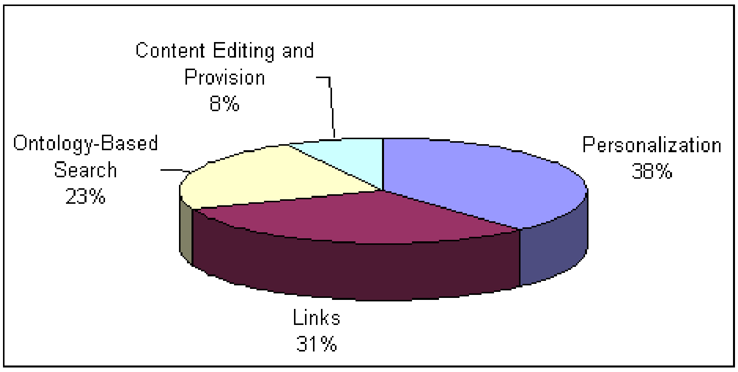


Figure 7-4 Pie chart represents the proportion of preferred functionalities of SEMPort

7.2.3 Phase 3 – An Overall Post-Questionnaire

To understand overall attitudes of the subjects to ECS CMWP and SEMPort, we used a post-questionnaire. We asked users to rate the usability of ECS CMWP and SEMPort. The results are represented as histograms as shown in Figure 7-5. Most of the subjects found the usability of ECS CMWP fair (2.9) and most of them found the usability of SEMPort useful (4.2). We also asked users to rate the usability of the tasks, and the results are illustrated in Figure 7-6. Generally, users found tasks fair on ECS CMWP (3.2) and useful on SEMPort (4.4). The rest of the results are presented in Table 7-7. The results showed that participants enjoyed using SEMPort more (4.2) compared to ECS CMWP (2.3). All of the subjects preferred to use SEMPort and 10% of them preferred to use ECS CMWP in the future.

7.2.4 Overall Comments on the Procedure

Overall, the two-phased experiment was completed by subjects in 40-90 minutes. During this time, the answers of the participants were noted and an observer recorded observations about the experiment.

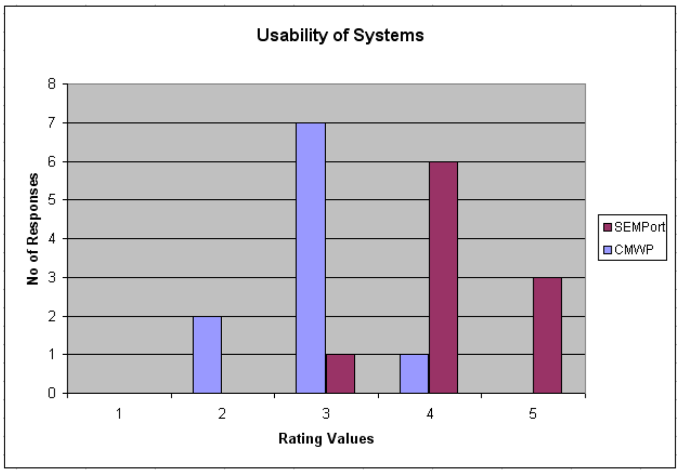


Figure 7-5 Usability of ECS CMWP and SEMPort

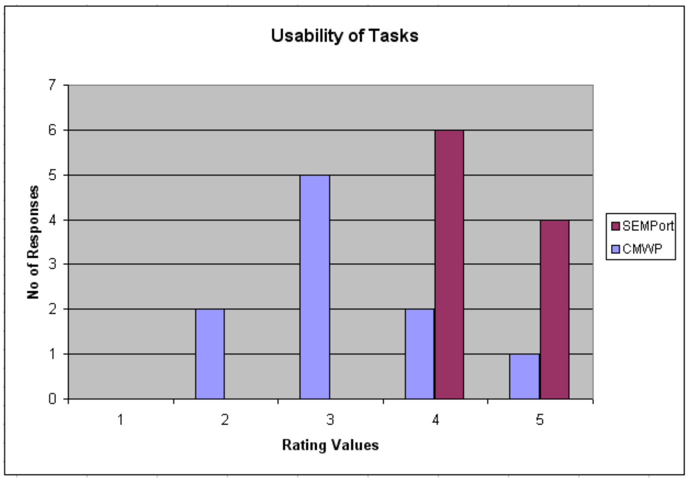


Figure 7-6 Usability of the tasks on ECS CMWP and SEMPort

Table 7-7: Overall questionnaire results on ECS CMWP and SEMPort

	ECS CMWP	**SEMPort**
Enjoyed using the system	2.3 (Mean)	4.2 (Mean)
Would continue using the system in the future	10% Yes	100% Yes

7.3 Summary of Evaluations

The structured review was performed on the initial version of SEMPort. The review provided guidelines for the improvement of the user interface and a number of usability problems were found. Before the user testing, the majority of the problems associated with "visibility of the system status" and "match between system and the real world" were fixed. The remaining problems were left to be fixed in the future. A number of lessons have been learned from the evaluation.

- Although the discount usability evaluation method will not find all of the usability problems, it is a quick and cost effective solution. In particular, the ten heuristics of Nielsen provided a sound basis for the evaluation.
- Performing evaluation one reviewer at a time, worked really well. The reviewers did not bias each other and different evaluators found different usability problems.
- Although some of the usability problems were fixed, the problems associated with the "visibility of the system status" and "match between the system and the real world" improved the look and feel of the interface. The use of descriptive titles for ontology classes and attributes especially solved the misunderstandings significantly. We had changed the titles of the ontology concepts and their attributes before user testing. During the user testing, the titles were understood by all of the participants.

In addition to the interface problems, reviewers made useful comments on some aspects of the portal. The ontology hierarchy was found to be an effective way of providing access to different concepts. Links were found to be extensive and very functional. Personalized homepages were found to be very helpful. The content editing/provision Web interface was found to be useful and easy to use.

On the other hand, in our opinion, we could have improved the structured review with more in-depth analysis. These improvements could be:

- Five evaluators can identify 75% of the usability problems according to the study (Nielsen and Molich, 1990). SEMPort could have been tested with five evaluators instead of four. Five evaluators could have identified more interface problems.
- Because of the limited time before empirical study, we did two iterative evaluations using the structured review. In the first round, with four evaluators and in the second round, with two evaluators. Doing three iterative evaluations with all evaluators could have helped to identify more interface problems because of the design-evaluate-design recycle.

A two-phased experiment was setup on ECS CMWP and SEMPort. The experiment is conducted with ten postgraduate students. In our opinion, it could have been more useful to test SEMPort on undergraduate students, since undergraduates use ECS

CMWP for selecting and following modules. In the first part, we compared ECS CMWP and SEMPort in terms of navigation by using a set of tasks. Note that for the purpose of this experiment SEMPort used the same information as ECS CMWP. The results of the experiments showed that participants performed tasks better by using SEMPort (98% satisfaction) compared to ECS CMWP (41% satisfaction). In addition, questionnaire ratings of the subjects confirmed these results. Subjects rated 4.4 out of 5 for satisfaction with semantic navigation, compared to a mean rating of 1.9 for the navigation of ECS CMWP. In addition, in SEMPort, the hyperlinks presented were found to be effective with an average rating of 4.5, compared to 2.9 for ECS CMWP.

In the second part, we analyzed the functionalities of SEMPort by using a set of tasks on ontology-based search, personalization, and content editing/provision. The results showed that the subjects were satisfied with ontology-based search with 4.4 mean, satisfied with personalization with 4.5 mean, and satisfied with content editing/provision with 4.4 mean. The results from the questionnaires were encouraging. We also made a questionnaire to find out which features were liked most by the subjects. The most preferred features were personalization (38%) and the additional links (31%), which gave us guidance for our future work. In addition, SEMPort was found to be useful and 100% of subjects wanted to continue to use SEMPort.

The work on SEMPort motivated us to develop a system for providing personalization and dynamic linking in broader contents, such as the WWW. In SEMPort, dynamic link creation and personalization was specific to one domain (portal domain) and we wanted to extend this to a broader area, such as to Web browsing. For this purpose, we have extended the standard Web browser to embed ontology-based links and to support AH during Web browsing, which we called SemWeB. With SemWeB, Web browsing can be enriched with ontology-based links and information can also be personalized to individual users. However, there are challenges to achieve these. For instance, which user characteristics can be used to provide Web-based personalization, which vocabularies can be used to support context-based link creation, also the system needed to be open-corpus and able to work on different domains. These motivations and the lessons we have learnt from SEMPort (i.e. ontology-based link creation and ontology-based user modelling), led us to develop SemWeB, which will be explained in next chapter.

8 SemWeB – A Personalized Semantic Web Browser

8.1 Motivations

Our work on SEMPort motivated us to develop a system to support hyperlinking and AH on different Web domains using Semantic Web technologies during browsing. Searching and browsing are two important information filtering activities on the Web. Usually, users use search engines for finding Web resources but this is only half of the story. When users follow a link from search results, they have to read and understand page content and in general they are not guided during browsing, which is a complex activity. It is our hypothesis that, browsing can be supported using Semantic Web technologies and AH methods. Semantic Web technologies provide powerful knowledge representation formalisms and inferencing mechanisms on the Web. Browsing can be enriched by the power of these technologies. Additionally, different users have different browsing needs and page content and hyperlinks should be adapted accordingly. AH is a solution, where personalization mechanisms adapt information to the needs of the users. Thus, our main design goal is to enrich browsing with semantic information (content and links), also generating and adapting data based on the information needs of the users.

Personalization is supported by many websites on the Web (e.g. Amazon, Google, and Yahoo). However, they are obstructive; users are required to log in to multiple websites and enter their personal information and preferences, and the profiles are different for each site. There is a need for generic user profiles and personalization architectures, which can achieve AH on diverse websites. Semantic Web technologies can again offer the solution to these problems. Ontology-based user profiles that are represented with a common agreed ontology are interoperable, and they can be easily extended with metadata. Additionally, user profiles should support the user's browsing for Web-based personalization. However, current user modelling standards (i.e. IEEE PAPI and IMS LIP) do not support this information.

Our research has been motivated by the needs as discussed above, also it is inspired by open hypermedia systems, particulary Microcosm (Fountain et al., 1990; Hall et al., 1996). The Microcosm system provided hyperlinking from all types of multimedia documents in an open corpus. It also provided a framework for building AHSs as discussed by (Hothi and Hall, 1997). In Microcosm, documents and link data are separated; *linkbases* are employed to store link information. This approach reminds us linked data. Linked data provides metadata about world objects and it is stored distributedly on the Web in a similar manner to the *distributed linkbases* of the Distributed Link Service (see chapter 2.1). Linked data can be used as the source of hyperlinking for context-based hyperlink generation on Web documents.

Related works to SemWeB are COHSE (Carr et al., 2001, Yesilada et al., 2008), KIM (Kiryakov et al., 2004), Magpie (Dzbor et al., 2003) and PowerMagpie (Gridinoc et al., 2008). All of these systems provide semantic hyperlinks on documents using ontologies. For instance, COHSE, KIM and Magpie utilize their own ontologies to annotate and generate ontology-based hyperlinks. In contrast, PowerMagpie recognizes linked data instances from a Web page using the Watson search engine and then it dereferences the linked data URIs for ontology-based link generation. Thus PowerMagpie does not require a pre-defined ontology. The differences between SemWeB and these systems are goal services (context-based link generation to relevant Web resources) and the support of AH. Recently COHSE provided ideas for personalization but these features have not been implemented (Yesilada et al., 2008).

We propose a novel personalized Semantic Web browser, which we call *SemWeB* (Şah et al., 2008), (Şah et al., 2008b) and (Şah et al., 2009). SemWeB is an AJAX and Javascript based browser extension, which performs IE and semantic annotation to interpret page content using linked data and provides semantic information and adaptation according to the user model. For adaptation, we propose a new behaviour-based and ontology-driven user model, in which users can add goals, interests and expertise data to their profiles from their browsers. By using a Web browser, users can surf on the Web as normal and with SemWeB this experience can be enriched with semantic information, links and personalization. In the rest of this chapter, we discuss the SemWeB system design, semantic annotation mechanism, the proposed user model, semantic linking approach, and employed AH methods and techniques. In addition, the reader is referred to the Appendix D for detailed walkthroughs of the use of SemWeB.

8.2 System Design

For understanding user interaction with the Web and to enable AH on different Web sites, we implemented SemWeB as a browser extension of the Mozilla Firefox Web browser. The system design of SemWeB is depicted in Figure 8-1. SemWeB extends the Web browser with a vertical sidebar. The sidebar is used for choosing ontologies, annotating pages with metadata, embedding hyperlinks and interacting with users. With a Web browser, users can surf the Web, in addition to this, with the SemWeB browser extension, ontological concepts within the Web page can be found and more information and links about them can be presented. To do this, Information Extraction (IE) is needed to extract conceptual instances from the page. For this purpose, we employed GATE, which is an open source platform for text processing tasks (Cunningham et al., 2002). We extended GATE with a lookup service as well as an annotation generation and storage unit. Web pages can be dynamically annotated on the user's request from the browser extension by the Information Extraction Service (IES). We assume that the user will choose an ontology for use within the browser. Depending on the selected ontology, different annotations are made to the same page.

Based on the selected ontology, ontological concepts are presented at the SemWeB sidebar. Users can highlight the found ontological instances on the Web page using these concepts. But first they need to annotate the Web page. Once, the page is annotated, the user can highlight concepts or can select the provided browsing goals from the SemWeB sidebar. When the user select a concept, then the SemWeB browser extension embeds icons (hyperlinks) to the Web page (dynamic semantic hyper-linking) as shown in Figure 8-2. Using these hyperlinks, users can request semantic information, which is then handled by the Semantic Linking Service (SLS), which receives a dereferenceable URI, a user ID, user goal(s) and lexicon of the resource as inputs and creates semantic information and links. SLS first dereferences the URI over the HTTP protocol and creates semantic hyperlinks and information. In this step, if the user is logged in, then information and links are personalized by adapting to the individual's current information needs based on the proposed user model by the adaptation module. For example, if the user requested a browsing goal, then adaptive links to Web resources on the Web are generated. Or, the created semantic hyperlinks are annotated with different visual cues according to the relatedness to the user profile using a novel semantic relatedness measure. In addition, based on expertise values, adaptive text (i.e. Wikipedia definition) or adaptive links (i.e. links to related Wikipedia pages) are created by the adaptation module. Finally, personalized information and links are returned to the client's browser and presented in a new Web page as illustrated in Figure 8-14.

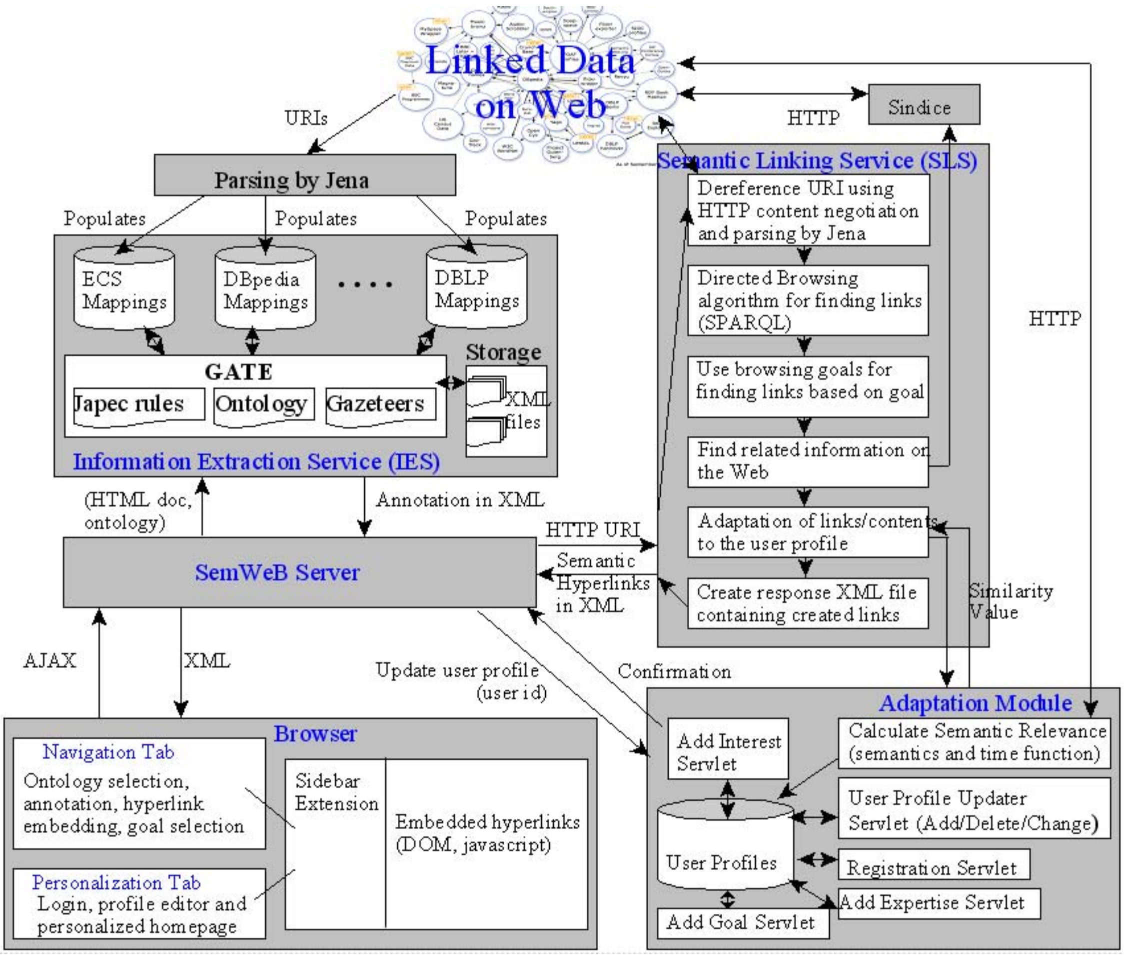


Figure 8-1 The system design of SemWeB

With walkthroughs, the interactions between the browser, different SemWeB modules and linked data Web can be explained in detail as follows: First the user needs to open

SemWeB sidebar extension and chooses an ontology for use with the browser. Then, the user requires to annotate the Web page. When the user requests annotation of a Web page, an Asynchronous XML (AJAX) request is sent to the IES module. IES loads gazetteers and JAPEC rules, and then annotates the Web page according to the selected ontology. This annotation is stored at the server-side in XML and sent back to the browser. Browser extension receives the data using XMLHttpRequest and uses Javascript functions and Document Object Model (DOM) to embed hyperlinks and semantic descriptors to the Web page at the browser. When user clicks onto embedded hyperlinks, then this request is sent to SLS. SLS first dereferences the requested URI from the linked data Web and retrieves its RDF description. Then, it uses directed browsing algorithm to create semantic links and invokes adaptation module for personalization if the user is logged in. The adaptation module queries the user database for obtaining the user profile and uses adaptation rules and algorithms for personalizing information (i.e. link annotation). Finally, SLS creates an XML response which contains the generated personalized links and information and then sent this data to the browser. This data is received using XMLHttpRequest and presented in a new Web page using DOM and Cascading Style Sheets (CSS). Users can also add interests, expertise values or browsing goals to their profiles from their browsers. Also, they can use a profile editor and access to their personalized homepages from their SemWeB browser extension. All of these updates are handled by the adaptation module.

8.3 SemWeB Browser Extension

SemWeB extends the Mozilla Web browser with a sidebar. The sidebar has two tabs: the navigation tab and the personalization tab.

The navigation tab is used to select a view on the page, by selecting an appropriate ontology as shown in Figure 8-2. Different ontologies provide different annotations on the page. For experimentation of SemWeB, we used the ECS ontology and instances (ECS Southampton, 2008). Additionally, we tested it on DBPedia (DBpedia, 2008) and DBLP (DBLP, 2008), showing it can be adapted to diverse ontologies. Once an ontology is selected, the user can annotate the Web page based on the selected view of the ontology from the navigation tab. To prevent too many hyperlinks being shown at one time, users are choose what kind of information they want to see. To do this, concepts from the selected ontology are presented in the navigation tab (Figure 8-2) and users can choose the concepts they are interested in. Once a user selects a concept, the instances of the selected class are highlighted and the SemWeB browser extension embeds icons next to the recognized instances on the Web page. To prevent existing hyperlinks being overriden, links are added next to the instance names. Subsequently, users can request semantic information and links by clicking the embedded icons.

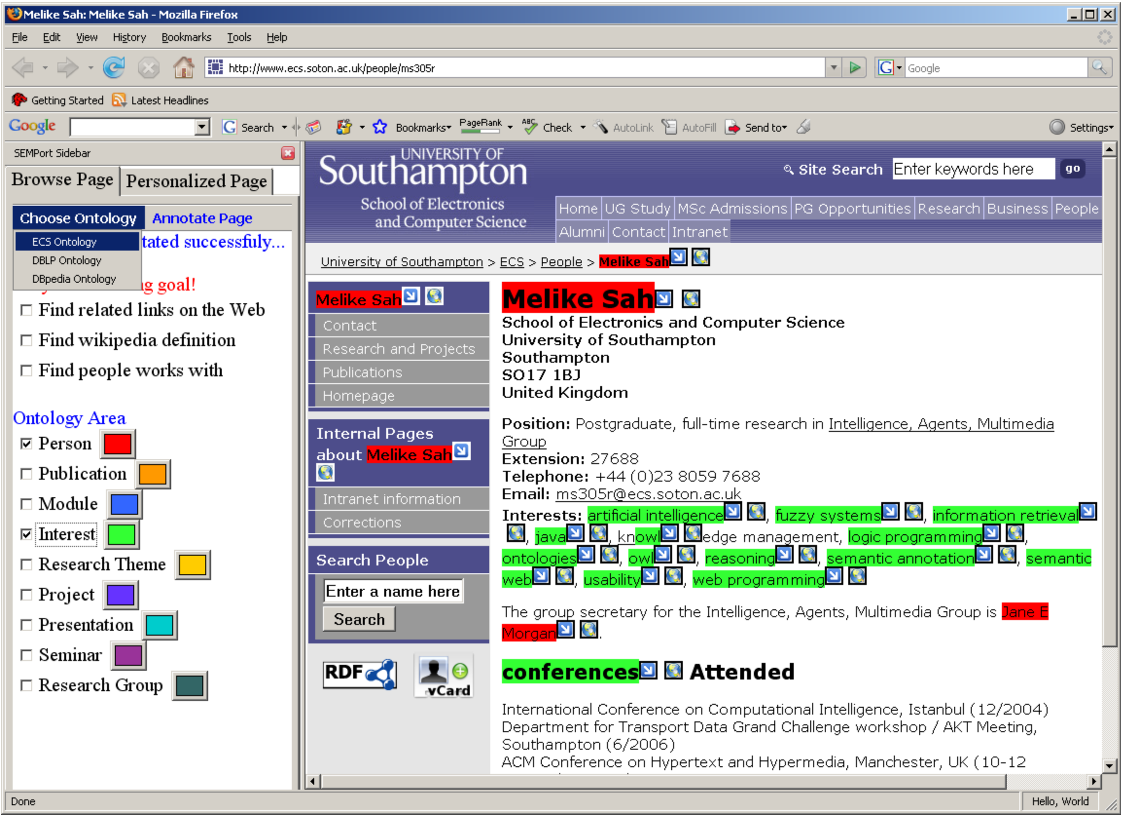
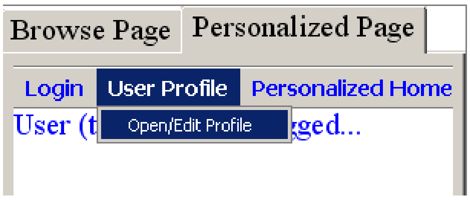


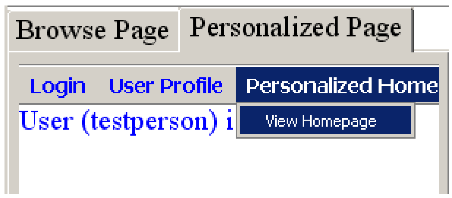
Figure 8-2 A screen shot of SemWeB with the added semantic links

When the user clicks on an embedded icon, the SemWeB server is responsible for finding semantic links and related information on the Semantic Web and the results are personalized according to the user model and presented adaptively in a new Web page. In addition, users can add semantic instances to their interests and expertise any time from their browsers. They can also insert browsing goals into their profiles from the navigation tab (Figure 8-2). Depending on the selected ontology, different sets of goals are provided, which will be discussed later in this chapter.

The personalization tab is used by users to login to their user profiles as shown in Figure 8-3 (a). Additionally, users can update their profiles from the profile editor (see Figure 8-3 (b)) and access to their personalized homepages (see Figure 8-3 (c)).

(a) Login interface from personalization tab





(b) Access to user profile (c) Access to personalized homepage

Figure 8-3 SemWeB sidebar extension – personalization tab

8.4 Semantic Annotation using SemWeB

One of the design aims of SemWeB is to use existing linked data for semantic linking, instead of creating and storing metadata in databases. For IE and semantic annotation, therefore we utilize linked data. In order to make use of linked data, first we choose a domain (e.g. ECS) and analyse it; identify HTTP URIs and their lexicons. Then, for semantic annotation, named entities from the Web pages have to be extracted and annotated using the underlying ontologies. Finally, annotated Web pages can be used to present semantic information and services depending on the preferred ontology.

For experimentation with SemWeB, we used the ECS ontology and associated instances (ECS Southampton, 2008). The ECS provides metadata about people, publications, modules, etc. within the School of Electronics and Computer Science in the University of Southampton and publishes this information as linked data. This metadata can be accessed over the Web using HTTP content negotiation. Thus, we do not hold a central database for metadata; instead we rely on decentralized metadata located on the Web. For IE, SemWeB uses ontologies and an ontology-driven lexicon based on a modified GATE framework (Cunningham et al. 2002).

GATE has been developed at the University of Sheffield as an open source text engineering architecture for extracting named entities from documents. It contains a complete set of information extraction components known as ANNIE (A Nearly New Information Extraction System). These components are reusable and called Processing Resources (PR). PRs are designed to produce GATE annotations from a corpus of source text, where a GATE annotation adds a set of features within span tags into the source text. GATE can process HTML, XML, SGML, email, rtf, plain text, PDF, Microsoft word formats. PRs are combined into pipelines, where each PR read the document and can access the GATE annotations created by previous PRs. PRs can be used individually or joined together or additional resources can be included. This open nature of GATE allows it to be used by variety of IE processes and other tasks. For this purpose, we decided to use GATE for IE and in the following section, the IE and semantic annotation process is explained in more detail.

8.4.1 Annotation Pipeline Used in SemWeB

The SemWeB annotation pipeline is built on the basic ANNIE pipeline supplied by GATE, which is shown in Figure 8-4. Currently, GATE version 4.0 is employed. We added some additional components and some changes were done to the existing components. The following changes were made to GATE:

- JAPEC is a pattern matching rule interface for GATE. We extended JAPEC with a lookup service, which is used to match a lexicon with a dereferenceable URI. In addition, new JAPEC rules were added to improve IE.
- An annotation generation and storage unit was added to GATE pipeline. This unit converts GATE annotations to XML based on the used ontology and store the created XML file at the server-side and return this annotation to browser.
- In the DBpedia domain, an *idf* (inverse document frequency) unit was added. It is used to filter important DBpedia instances based on their occurrences in the whole DBpedia corpus and will be discussed in chapter 9.

Annotation pipeline components are explained in detail as follows:

Tokeniser: The first component is tokeniser. It is used to split input document(s) into simple tokens, such as words, numbers, and symbols. Currently, the default English tokeniser from GATE's ANNIE pipeline is employed. The tokeniser can be supplied for different applications and text types without modification.

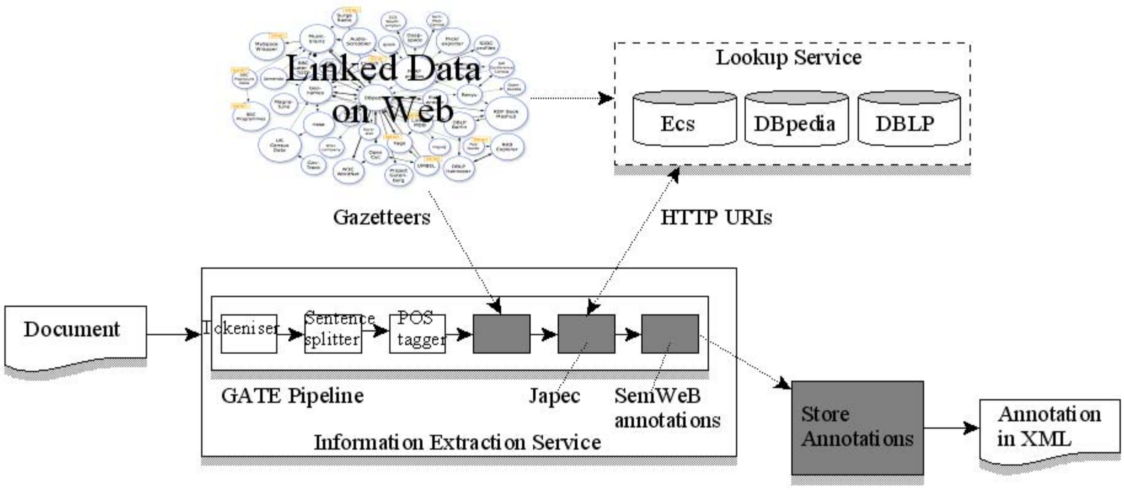


Figure 8-4 Annotation pipeline architecture

Sentence Splitter: The sentence splitter tries to break texts into sentences, which is required for part-of-speech tagger. Again, the ANNIE sentence splitter was used in the pipeline. It is application and domain independent.

POS Tagger: The part-of-speech tagger is a general purpose tagger which is used to produce tags for each token with a part of speech. At the end of this process, all tokens are tagged with most likely candidate of part of speech. Again, the ANNIE POS tagger was used. Tagger is domain and application independent.

Gazetteers: The next component in the pipeline is gazetteers. Gazetteers consist of list of names occurring in text, such as list of city names. Gazetteers are used to match against tokens identified from the document. In addition, in GATE, every gazetteer has a major type and minor type. If a token is matched with a name from a gazetteer, the document is annotated with features specifying the major and minor types. On the other hand, ANNIE came up with list of gazetteers, for instance city names, organizations, days of the week, etc. During the semantic annotation, our aim is to map a lexicon to a dereferenceable URI, so that we can look up it later. Therefore, we did not use built in ANNIE gazetteers. Instead, all RDF files on the ECS pages are crawled and gazetteers were created based on the found URIs and their associated lexicons. ECS provides an interface[51], where all created RDF files can be queried by using time stamp=0. We used this interface to find dereferenceable URIs, we then dereferenced and parsed these URIs using Jena and searched for lexicons using SPARQL queries. We have used

[51] http://id.ecs.soton.ac.uk/list/ [last accessed, 10/6/2008]

different annotation properties for creating lexicons (see Table 8-1) and different types of gazetteers were produced depending on different class types. Finally, the created gazetteers were added to the pipeline and they were mapped to an ontology class. Additionally, lexicons and associated URIs were stored to a database called *mapping database*, so that later lexicons can be match to a valid URI or vice versa.

It is also noted that gazetteer generation requires extra processing and a potential bottleneck in the system due to crawling required. However, almost all semantic annotation tools or approaches need pre-processing to analyze lexicons of the domain since prior knowledge is essential as discussed in chapter 5.4. In addition, in GATE, large lexicon corporas makes the system to slow down since the time needed to load gazetteers. This can be alleviated by using database gazetteers. For instance, lexicons can be stored to databases and indexed for faster processing. The improvements of the scalability of the semantic annotation will be carried out in future work. More detailed discussion of the scalability of semantic annotation is discussed in chapter 9.1.3.2.

Table 8-1: Annotation properties that is used in ECS

Gazetteer Name	**Annotation Property Used**	**Major Type**
Person_Full_Name.lst	ecs[52]:hasFullName	Person_Full_Name
Surname.lst	ecs:hasFamilyName	Surname
Publication.lst	ecs:hasTitle	Publication
Project.lst	ecs:hasName	Project
Module.lst	ecs:hasModuleTitle	Module
ModuleShort.lst	ecs:hasModuleCode	ModuleShort
Interest.lst	ecs:hasName	Interest
Degree.lst	ecs:hasName	Degree
Presentation.lst	ecs:hasName	Presentation
Cohort.lst	ecs:hasName	Cohort
Theme.lst	rdfs:label	Theme
Research_Group.lst	ecs:hasName	ResearchGroup

[52] http://rdf.ecs.soton.ac.uk/ontology/ecs# [last accessed, 13/6/2008]

Semantic Tagger (Named Entity Transducer): In ANNIE, the semantic tagger consists of hand-made written rules in the JAPE (Java Annotations Pattern Engine) language (Cunningham et al., 2000). JAPE describes patterns to match against multi-token sequences occurred in text and annotations are generated as an outcome. Patterns (action rules) are based on a set of regular expressions that are defined by JAPE grammar and the set of all patterns defines a finite-state transducer. Patterns can be described by a specific text string or by annotations previously created by other GATE modules, such as a tokeniser or gazetteer. In addition, rules can be prioritised to prevent multiple annotations on same text string. Finally, rules are executed sequentially and annotations are added to the original source text using "<span>" tags with annotation features. For optimizing the performance of the JAPE transducer, a JAPE-to-Java compiler[53] (JAPEC) was developed. It was purposefully implemented to achieve better performances and they claim that JAPEC is 2 to 5 times faster than JAPE. JAPEC also utilizes the JAPE grammar and allows pattern matching over the annotation set and the document text. In SemWeB, we employed the JAPEC transducer for semantic tagging. We created new JAPEC rules based on the gazetteer tokens we provided in the previous step. In addition, to match people names that can occur in different formats (i.e. Wendy Hall, W. Hall, Hall, W.), new pattern matching rules were generated as shown in Figure 8-5.

In order to find the URI of the recognized named entity, we extended JAPEC with a lookup service, which is a Java class connected to the mapping database and that searches URIs based on the provided lexicon. The JAPEC binary source code was extended with the lookup service and we can invoke this service from JAPEC rules directly as shown in line 19 in Figure 8-5. For each recognized named entity a URI must be found otherwise annotation for that entity fails. After executing the JAPEC transducer, the document is processed against patterns and annotations are added to the document based on the actions provided in the JAPEC rules. During the semantic annotation process, a specific name was given to the annotated texts, for instance "Mention", so that later we can identify properties added by GATE (see line 28 in Figure 8-5). In addition, the following features were added to the semantic annotation (see lines 25, 26 in Figure 8-5): class name (ontology class that the instance belongs to) and URI mapping (URI of the the recognized instance). For example, assume in the

[53] http://www.ontotext.com/gate/japec.html [last accessed: 13/6/2008]

source document the text “Wendy Hall” was identified. Then, the following annotation would be added to the document after running the JAPEC transducer as shown in Figure 8-6.

```
Rule:PersonName2
Priority:20    //Rule Priority
(({Lookup.majorType==Surname}):surname //Pattern to be match
({Token.string == ","})
((INITIALS)+):initials) --> { //actions to be taken
AnnotationSet mentionSet1=(gate.AnnotationSet)bindings.get("surname");
Annotation mentionAnn=(Annotation)((AnnotationSet)bindings.get("surname")).
iterator().next();
AnnotationSet mentionSet3=(gate.AnnotationSet)bindings.get("initials");
Annotation mentionAnn3=(Annotation)((AnnotationSet)bindings.get("initials")).iterator().next();
String anchor=""; String mapping=""; //anchor is recognized text, mapping is URI
try{
anchor=(doc.getContent().getContent(mentionAnn.getStartNode().getOffset(),
mentionAnn3.getEndNode().getOffset()).toString());
} catch(InvalidOffsetException io){}
//Invoking Lookup Service
com.ontotext.gate.japec.LookupService lookup=new com.ontotext.gate.japec.LookupService();
if(anchor.equals("")){}
else { try {
mapping=lookup.getMapping("Person",anchor); //obtaining URI of the recognized lexicon
}catch(Exception mapping_exception){System.out.println(mapping_exception.toString());}
if (mapping=="" || mapping=="empty" || mapping==null){}
else {
//annotation features to be added
FeatureMap features=Factory.newFeatureMap();
features.put("class", "Person");
features.put("mapping", mapping);
//add annotations to the document
annotations.add(mentionSet1.firstNode(), mentionSet3.lastNode(),"Mention", features);
//remove previously added annotations
annotations.removeAll(mentionSet1);
annotations.removeAll(mentionSet3);
} } }
```

Figure 8-5 An example JAPEC rule used to recognize person names

```
<span class="Person"
mapping="http://id.ecs.soton.ac.uk/person/1650">Wendy Hall</span>
```

Figure 8-6 An example GATE annotation added to the document

Annotation Generation and Storage: A new component was added to convert GATE annotations to SemWeB annotations. It searches for tokens containing “Mention”, which means a lexicon from the ontological instances is mentioned in the text. Then, “class” and “mapping” features are obtained and then those features are converted to XML. The same text may appear in more than one place in a Web page. To prevent multiple annotations of the same objects, duplicates are removed. The following figure shows how previously created GATE annotation in Figure 8-6 is presented in XML.

```xml
<?xml version="1.0" ?>
<message>
  <Person>
      <value>Wendy Hall</value>
      <mapping>http://id.ecs.soton.ac.uk/person/1650</mapping>
 </Person>
</message>
```

Figure 8-7 An example SemWeB annotation

In SemWeB, the semantic annotation is performed on the user's demand. Therefore, to prevent undesired delays during the semantic annotation, the created XML annotation of the Web page is stored to the server-side. In this way, if a page has been previously annotated the annotation is returned to the client's browser directly without delay. If it does not exist, the page is dynamically annotated at run-time. The annotation time-line between browser and the server is illustrated in Figure 8-8.

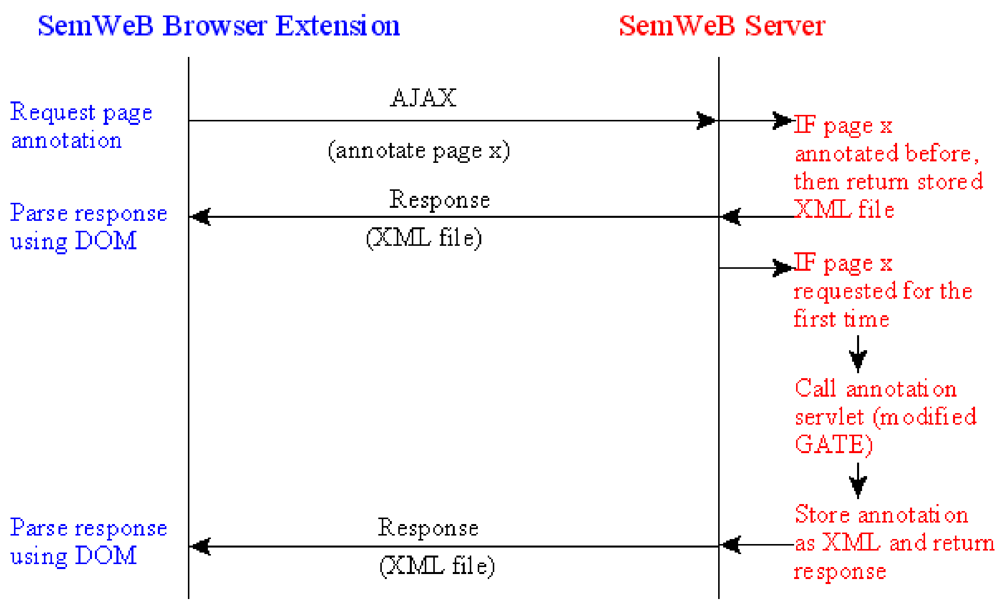


Figure 8-8 An illustration of annotation time-line

8.4.2 Creating and Running Annotation Pipeline

The pipeline can be run either via the GATE GUI or as a standalone service provided through an Information Extraction Service (IES). To use the GATE GUI for semantic annotation, the user needs to set up and initialize PRs as described above and then feed documents into pipeline.

In SemWeB, users can request the annotation of a Web page from their browsers using Asynchronous Javascript and XML (AJAX). With AJAX, the browser retrieves data from the server asynchronously without interfering with user browsing. As a result users are not interrupted while waiting for the response. The annotation request is performed by the IES at the server. The IES is deployed as a servlet hosted within a servlet engine (e.g. Tomcat). From the user's browser, a set of arguments are sent to the IES:

- The URL of the Web page (document), which will be annotated.
- The ontology name. The ontology domain where instance terms will be used in SemWeB annotations.

The IES first loads the document. Then depending on the ontology being used, corresponding gazetteers and JAPEC rules are deployed and PRs are applied in the sequence described above. After SemWeB annotations are created and stored, the annotation is sent back to the client's browser in XML as shown in Figure 8-8.

8.4.3 Handling Semantic Annotations by SemWeB Browser Extension

In our approach, users use their browsers as usual for surfing and navigating on the Web. Semantic information and linking is not provided unless users request them from the SemWeB browser extension. Thus, users first need to open the SemWeB sidebar and select an ontology (the preferred domain, where information is going to be added). Then they can request semantic annotation of the current Web page. As we discussed in the previous section, the IES performs the semantic annotation and the results are sent back to the SemWeB browser extension in XML.

The SemWeB sidebar extension retrieves the annotation using the XMLHttpRequest object and handles it using the HTML Document Object Model (DOM) (W3C, 2003). We prefer not to embed all annotated semantic instances in the original Web page immediately, because too many hyperlinks can cause disorientation to the user's browsing. Instead, the hyperlinks are added to the Web page according to the user's choices. For instance, users can choose ontology classes and the selected class's instances are highlighted in the Web page using HTML DOM regular expressions. This is achieved by matching lexicons of the class instances in the page content by applying regular expression matching technique. In this step, the SemWeB browser extension adds two icons (hyperlinks) to the Web page by employing HTML DOM and Javascript functions (i.e. "<span>" tags are added): the first is a link to the recognized instance and the second is a request to the SemWeB server for showing semantic information and hyperlinks about the instance. In Figure 8-9, added span tags are shown. For instance, when the user clicks on the first icon, the URI (http://id.ecs.soton.ac.uk/person/1650) is dereferenced by the browser and the ECS server redirects this URI to its HTML presentation (http://www.ecs.soton.ac.uk/people/ms305r). When the user clicks on the second icon, which we call the *explore icon*, this makes a request to the SLS at the server-side and it will show semantic information and links according to the URI provided. This service is responsible for creating semantic information and also adapts those data according to the user model, which will be explained in section 8.4.

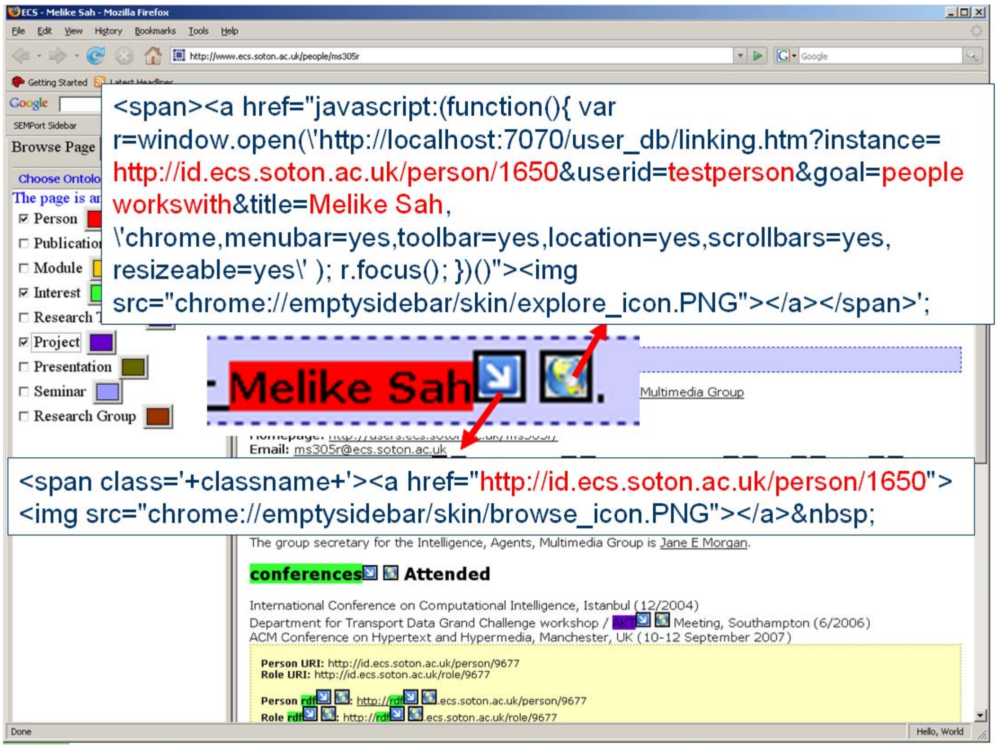


Figure 8-9 Added span tags by SemWeB browser extension

With our approach of embedding links using DOM, we are injecting new code into the user's copy of the Web page at the browser (not changing original Web page) and it will not break the original page. However, this approach may create DOM-based cross-side scripting volunerability at the local copy of the page at the browser. It should be also noted that there are possible modes of failure while embedding semantic annotations using the SemWeB browser extension. For instance, embedding annotations may fail if another AJAX application is rewriting the page at the same time. Besides, when too many annotations (the number of annotations depends on the cache memory size of the browser and size of the document) are tried to be embedded to the page at one time, the browser may give an error message of “excess scripting and out of memory”. To overcome this problem, we do not embed all annotations at once, but according to the choices of the user. On the other hand, another problem is automatic refreshing. If the annotated Web page has a Javascript function that refreshes the contents of the page within certain time periods, then embedded annotations will be erased, which is a potential problem to our approach.

8.5 The Proposed User Model

Before explaining how SemWeB creates ontology-based links and information and adapts this information to the needs of different users, it is necessary to first discuss the proposed user model. User modelling standards, such as IEEE PAPI (IEEE PAPI, 2008), IMS LIP (IMS LIP, 2008) and eduPerson (eduPerson, 2007) are widely accepted and utilized in many applications. These standards mainly use the learners’ knowledge and previous experiences for the adaptation. Although they can be applied to any domain, it is not suitable for Web-based Information Retrieval (IR) applications or the adaptive presentation of Web content: In IR-based applications, mainly user’s preferences, interests and goals are utilized for adaptation. Additionally, typical users are not willing to enter very detailed information to a Web site, such as the information required in IEEE PAPI, IMS LIP and eduPerson. As a result, the user model should be as lightweight as possible but should contain sufficient information about users’ preferences in Web-based IR applications. In addition, existing user models do not support the user’s browsing. As a result, we decided to create a new user model, which models the user’s browsing needs, preferences, interests and goals. The user model is also designed to be lightweight and can be applied to different domains (application-independent).

In order to support user browsing, we first need to understand their browsing behaviours. We thus analyzed existing research in the field of browsing behaviours.

8.5.1 Analysis of Browsing Behaviours

Browsing is a complex activity and its nature is not understood well. According to Bawden the activity of browsing can be categorized into three groups: purposive browsing, exploratory browsing and capricious browsing (Bawden, 1986). Bawden's browsing categories can be summarized as follows. In purposive browsing, users are looking for a definite piece of information (i.e. user has specific information in mind while browsing). Exploratory browsing is deliberately searching for inspiration (i.e. user does not have a specific goal but are seeking information motivated from his/her interests). Capricious browsing is randomly examining material without a defined goal.

Cove and Walsh, also divide browsing into three categories: search browsing, general purpose browsing and serendipity browsing (Cove and Walsh, 1988). We can outline Cove and Walsh's browsing types as follows. In search browsing, users are searching for a defined piece of information. General purpose browsing is looking for items of interest (i.e. browsing is directed by interests) and serendipity browsing is random and unstructured.

Based on these definitions, we can say that browsing tends to be used in three broad senses: a purposeful activity (directed from goals), searching for inspiration (semi-directed from interests) and capricious behaviour (undirected). In our opinion, user profiles should contain such information for supporting the browsing of users and we model this browsing behaviour in our proposed user model.

8.5.2 The Proposed User Model Ontology

We developed a new behaviour-based and ontology-driven user model, which we called the *user model ontology*. The user model ontology can be applied to different domains and it is published in (Şah et al., 2008; Şah et al., 2008b). The user model ontology is created using OWL-Lite by the Protégé ontology editor. All concepts of the user model ontology is shown in Figure A-11 in Appendix A, hierarchical relationships of classes are shown in Figure A-12 in Appendix A and the RDF/XML syntax of the user model ontology is attached to Appendix B. In our model, we currently use seven categories: identification, preference, security, *browsing goal, interest, expertise* and *browsing behaviour* (our main contributions are in italic). In future work, the user model could be extended with more information, such as portfolio. The new concepts introduced by our model are shown in more detail in Figure 8-10.

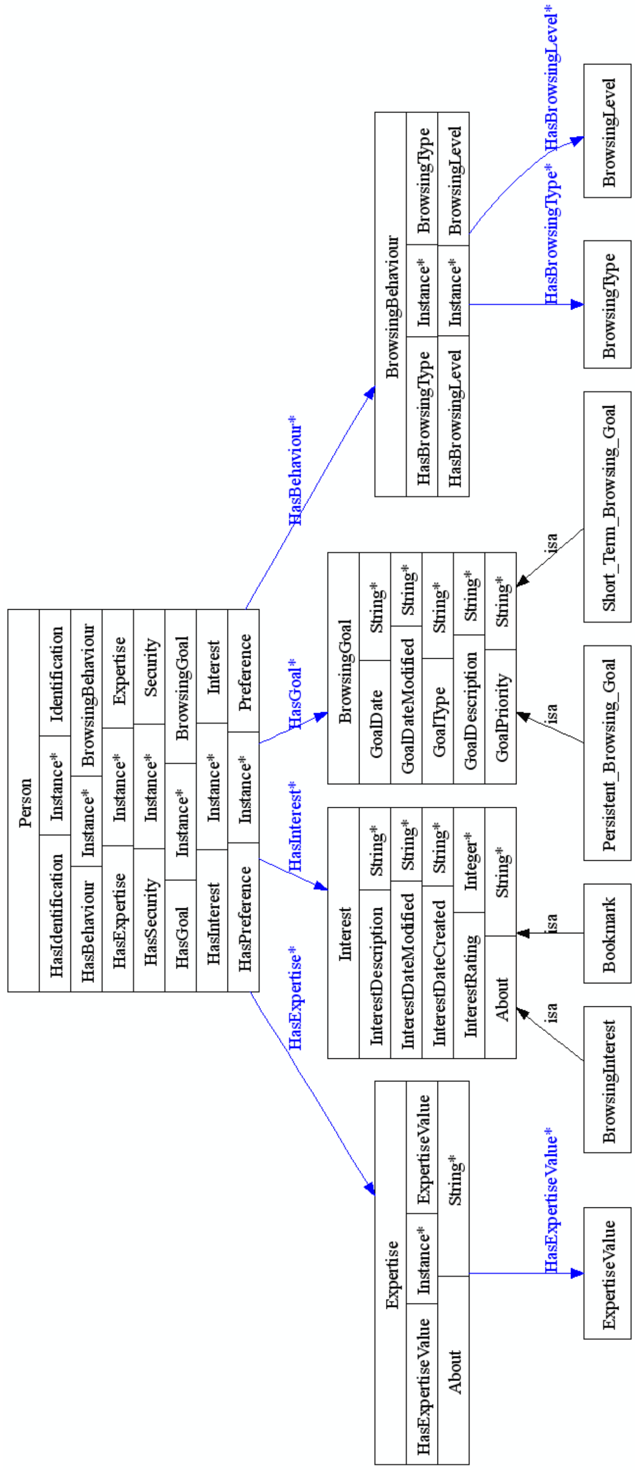


Figure 8-10 New concepts that are introduced by the proposed user model ontology (represented using Protégé OntoViz tool)

8.5.2.1 Concepts Used in the User Model Ontology

Identification: The *identification concept* contains personal information about users. For instance, name, surname, address, etc. It is a simplified version of the "Identification" category of the IMS LIP and the "Personal Information" category of the IEEE PAPI.

Preference: In the IEEE PAPI and IMS LIP, different categories are used to represent language and accessibility preferences of a user. In our model, *preference concept* contains layout, colour scheme and language preferences of the user.

Security: The *security concept* contains username and password.

Browsing Goal: We introduced the *browsing goal concept*, which represents the browsing aims of the user and is divided into two sub-concepts: *short-term browsing goal* and *persistent browsing goal*. The short-term browsing goals indicate the current information needs of the user (i.e. the goals of the user in a browsing session). The persistent browsing goals are the long-term goals of the user, which are motivated from long-term interests. For example a user interested in politics, probably likes politic related pages. Also, the browsing goal concept has five properties (see Figure 8-10): goal date, goal date modified, goal type, goal description and goal priority which are explained in Table 8-2. In SemWeB, browsing goals are automatically provided based on the selected ontology. Users are only required to select appropriate browsing goals from the SemWeB sidebar. Currently, we tested our system on the ECS, DBpedia and DBLP. In chapter 8.7, these services will be explained in more detail.

Interest: IMS LIP, IEEE PAPI and EduPerson represent interest as recreational activities or hobbies of learners. However in a Web environment, interest should indicate users focus on the Web (i.e. which pages they like, which concepts that are interested in, etc.) Therefore, our *interest concept* represents the browsing interests of users that can be understood from bookmarked pages and accessed semantic concepts and is divided into *bookmark* (interest to a webpage) and *browsing interest* (interest to a semantic concept). In addition, the interest category has five properties (see Figure 8-10): interest description, interest date created, interest date modified, interest rating (it can take three integer values: 1 represents low interest, 2 represents medium interest and 3 represents high interest) and about (the URI of the semantic instance that the user has interest to). These properties are explained in Table 8-2. In SemWeB, users can explicitly indicate their interests in semantic concepts (i.e. a Publication, Person, Location, etc.) recognized on the Web page from their browsers.

Currently, interest values can take discrete values in the user model, such as 1, 2 and 3. However, in future work, these values can be represented with continous fuzzy time series (Şah and Degtiarev, 2005). For instance, from the explicit user feedback or interactions with the browser, a fuzzy value, such as *very low, low, medium, high* and *very high* (which is a degree between 0 and 1) can be assigned to each interest at certain time intervals (note that these values can be calculated by the system based on previous records). Then, according to the trend, change or relationships between different fuzzy time series, the user's interest value (a degree between 0 or 1) for a certain instance can be predicted.

Expertise: According to the cognitive study of (Carmel et al., 1992), information should be presented differently to users with different expertise. For instance, this study shows that a novice user requires referential links to related information (i.e. related links and explanations). On the other hand, an area-expert needsarea-expert needs more detailed information about the same subject. To model this, we introduced *expertise concept* which represents the expertise of users for a semantic instance. It contains two properties (see Figure 8-10): has expertise value (can take novice, intermediate or expert values) and about (URI of the object that user specifies expertise). In SemWeB, users can explicitly enter expertise values to the semantic instances recognized on the Web page. Currently, expertise values can take discrete values, such as novice, intermediate and expert. As with interest values, expertise values can also be represented with fuzzy time series using fuzzy values.

Browsing BehaviorBehaviour: In order to implicitly understand the browsing activities of users, we introduced *browsing behaviour concept*. The browsing behaviour has *browsing_level* and *browsing_type* properties. Browsing_level (*very active, active, passive, inactiveand inactive*) is the number of clicks made by a user in a browsing session.

According to (Bawden, 1986) and (Cove and Walsh, 1988), different browsing strategies exist and we use browsing_type concept to represent this. In our approach, the browsing strategy of the user is implicitly understood from interactions they made in a browsing session without explicit feedback. For instance, if the user is using browsing goal services, then browsing might be directed by goals. The browsing strategy of the user can also be obtained by explicitly asking questions to user. However, we prefer to understand this from actions of the user according to the principles of (Bawden, 1986) and (Cove and Walsh, 1988). We use this philosophy as the basis of determining browsing_type values, such as *directed, semi-directed* or *un-directed* and these values are assigned according to the following conditions:

- When the user has a short-term browsing goal, it is assumed that user is looking for a defined piece of information and browsing_type is set to "*directed*". This accounts for the perception of search browsing or purposive browsing.
- When the user has a browsing interest or has bookmarked a particular Web page, it is assumed that the user is looking for items of interest and browsing_type is set to "*semi-directed*". This accounts for the perception of general purposive browsing or explanatory browsing.
- When the user does not have short-term browsing goals or browsing interests, browsing_type is set to "*undirected*". This accounts for the perception of serendipity browsing or capricious browsing, which is random browsing without goals or interests.

It should be noted that our interpretation of browsing_type makes the browsing behaviour discrete rather than possibly continous. In future, overlapping and continous browsing_type values can be design to represent browsing strategy. For example, if the user has a browsing goal and browsing interest at the same time, then browsing_type can be assigned based on continous behaviour rather than a discrete value.

8.5.2.2 Set of Properties Used in the User Model Ontology

Table 8.2 summaries the set of properties used in the user model ontology.

Table 8-2: A set of properties used in the user model ontology

Property Name	**Definition**
About	Defines the URI of a interest or expertise in Browsing Interest and Expertise concepts (i.e. http://id.ecs.soton.ac.uk/interest/owl)

AddressType	Type of the address (i.e. home, work, etc.)
City	City where the user resides
ColorSchemePreference	The preferred color scheme for the presentation
Country	Country where the user resides
Email	Email address of the user
GoalDate	The date when the goal created (i.e. 12-9-2008)
GoalDateModified	The date when the goal last modified (i.e. 12-9-2008)
GoalDescription	Description of the goal
GoalPriority	The importance of the goal to the user (can take values [1-3])
GoalType	Type of the service. (i.e. goal service provided by SemWeB)
HasAddress	Defines the relationship between Person and Address concepts
HasBehaviour	Defines the relationship between Person and Behaviour concepts
HasBrowsingLevel	Defines the browsing level of the user (i.e. very active)
HasBrowsingType	Defines the browsing type of the user (i.e. undirected)
HasExpertise	Defines the relationship between Person and Expertise concepts
HasExpertiseValue	Indicates the expertise value of the user (i.e. novice)
HasGoal	Defines the relationship between Person and Goal concepts
HasIdentification	Defines the relationship between Person and Identification
HasInterest	Defines the relationship between Person and Interest concepts
HasLanguage	Defines the preferred language of the user
HasPreference	Defines the relationship between Person and Preference concepts
HasSecurity	Defines the relationship between Person and Security concepts
InterestDateCreated	The date when the interest created (i.e. 1-9-2008)
InterestDateModified	The date when the interest last modified (i.e. 1-9-2008)
InterestDescription	Description of the interest
InterestRating	Indicates the importance of the interest to the user (i.e. medium)
LanguagePreference	The preferred language of the user
LayoutPreference	The preferred layout of the user
Name	The name of the user
Password	Security password of the user
Postcode	Postcode of the address
Street	Street address of the user
UserName	User name for log in to SemWeB

8.5.2.3 Instances Created within the User Model Ontology

A list of created instances within the user model ontology is shown in Table 8.3.

Table 8-3: Instances created within the user model ontology

Concept	Instances
BrowsingLevel	inactive, passive, active, very active
BrowsingType	undirected, semi-directed, directed
ExpertiseValue	novice, intermediate, expert

8.5.2.4 The Semantic Metadata of the User Model Ontology

The SemWeB server uses explicitly added information (i.e. interests, expertise, personal data, etc.) and implicitly understood information (i.e. browsing strategies of the user) and creates the RDF description of the user profile according to the user model ontology. An example user profile is presented as a directed labelled graph in Figure 8-11. The same profile is presented in RDF/XML syntax in Figure 8-12.

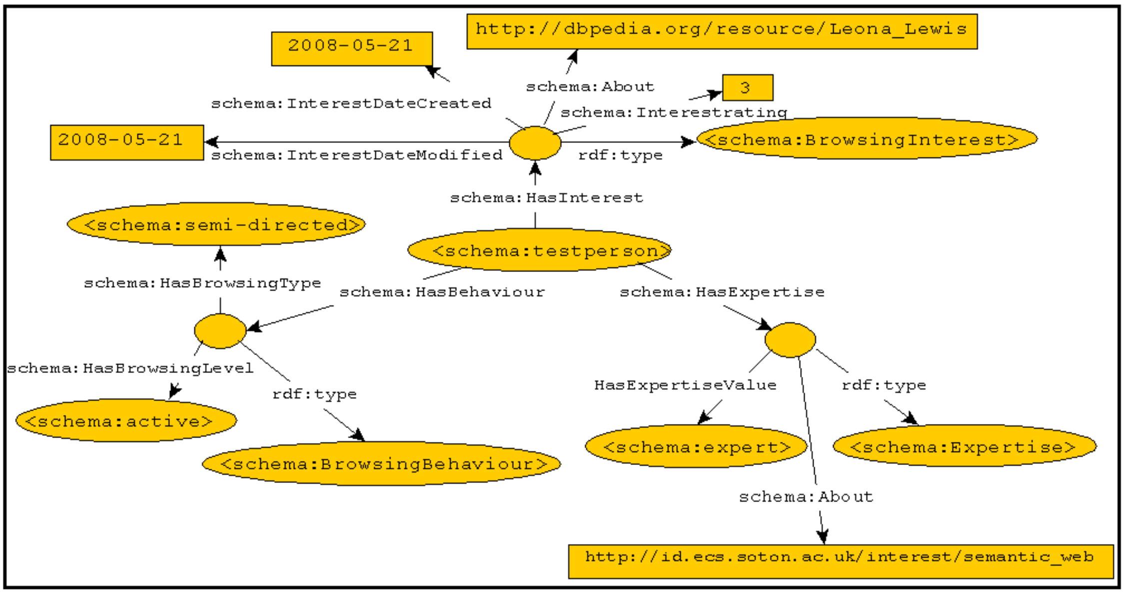


Figure 8-11 An example user profile represented as a directed labelled graph

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:schema="http://localhost:7070/user_db/user_schema.owl#"
    xmlns="http://localhost:7070/user_db/user_schema.owl#">
<rdf:Description rdf:about="#testperson">
  <schema:HasInterest>
    <schema:BrowsingInterest>
      <schema:About> http://dbpedia.org/resource/Leona_Lewis</schema:About>
      <schema:InterestRating>3</schema:InterestRating>
      <schema:InterestDateCreated>2008-05-21</schema:InterestDateCreated>
      <schema:InterestDateModified>2008-05-21</schema:InterestDateModified>
    </schema:BrowsingInterest>
  </schema:HasInterest>
  <schema:HasExpertise>
    <schema:Expertise>
     <schema:About> http://id.ecs.soton.ac.uk/interest/semantic_web</schema:About>
     <schema:HasExpertiseValue rdf:resource="#expert"/>
    </schema:Expertise>
  </schema:HasExpertise>
  <schema:HasBehaviour>
     <BrowsingBehaviour>
      <HasBrowsingLevel rdf:resource="#active"/>
      <HasBrowsingType rdf:resource="#semi-directed"/>
     </BrowsingBehaviour>
  </schema:HasBehaviour>
</rdf:RDF>
```

Figure 8-12 An example user profile using the proposed user model is presented in RDF/XML syntax

8.6 User Modelling

In the previous section, we explained the proposed user model which we developed for personalization. In order to start personalization process, users are required to register and login to SemWeB. Then, user profiles can be explicitly and implicitly updated.

8.6.1 Registering and Log in to SemWeB

When users select the “Register” menu item from the personalization tab of the SemWeB sidebar, a registration page appears (see Figure 8-13). To make things easier, this interface is simple and less information is required compared to IEEE PAPI, IMS LIP or EduPerson, for example identification (i.e. name, address), security (i.e. password) and preference (i.e. language, colour scheme, etc.) information is requested. Later, interests, expertise and goals can be entered by users any time they are browsing. When this form is submitted, it is handled by the Registration Service (RS) which is a servlet hosted within the Web server. The RS creates an RDF description of the user from the provided information according to the proposed user model using the Jena API. Then, it permanently stores the created user profile into the *user profiles* database, which is a Jena triple store. Currently all user profiles are kept at this central database. Users can log in to SemWeB immediately right after completing registration process from the "Log in" menu item provided in the personalization tab.

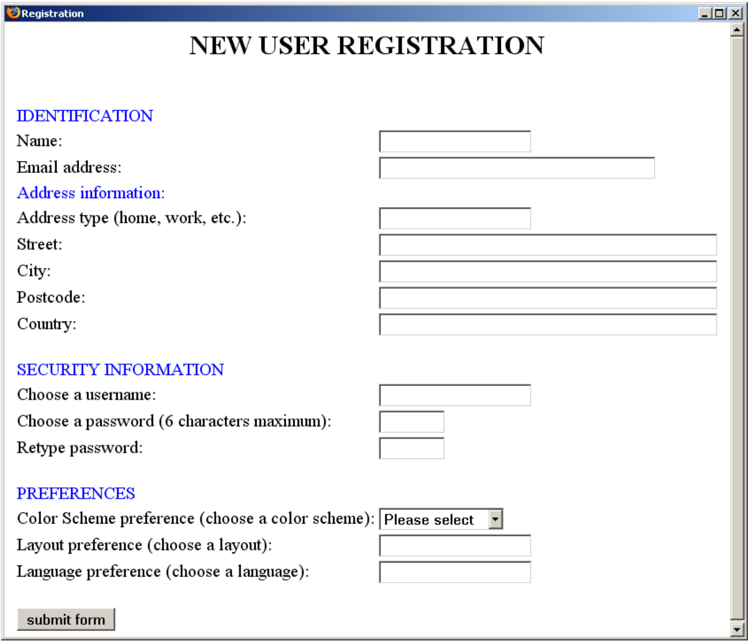


Figure 8-13 A screen shot of a registration page

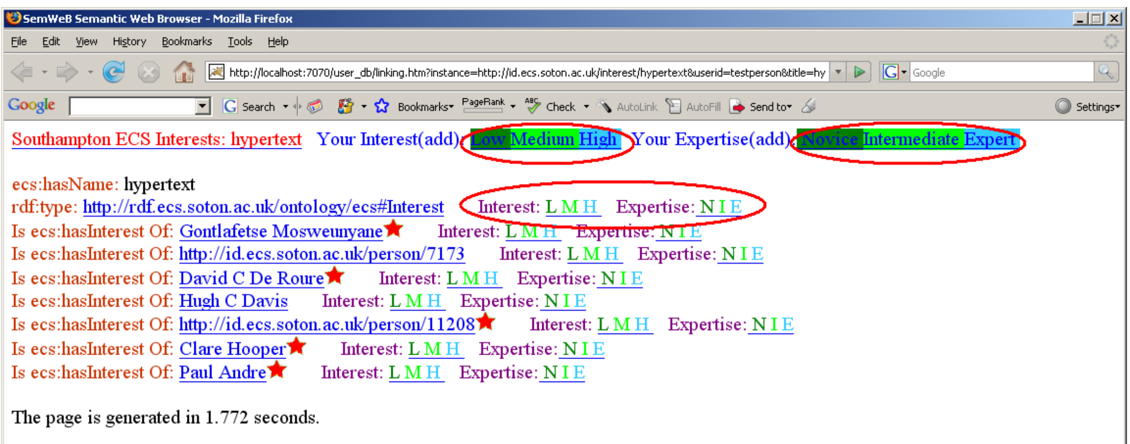
8.6.2 Adding Interest and Expertise Values to the Semantic Instances

Once the user is logged in to SemWeB, all presented semantic instances provided by the SLS are visualized with the addition of interest rating and expertise value buttons as shown in Figure 8-14 (a). Users can click on these buttons to enter interest and expertise values to the semantic instances.

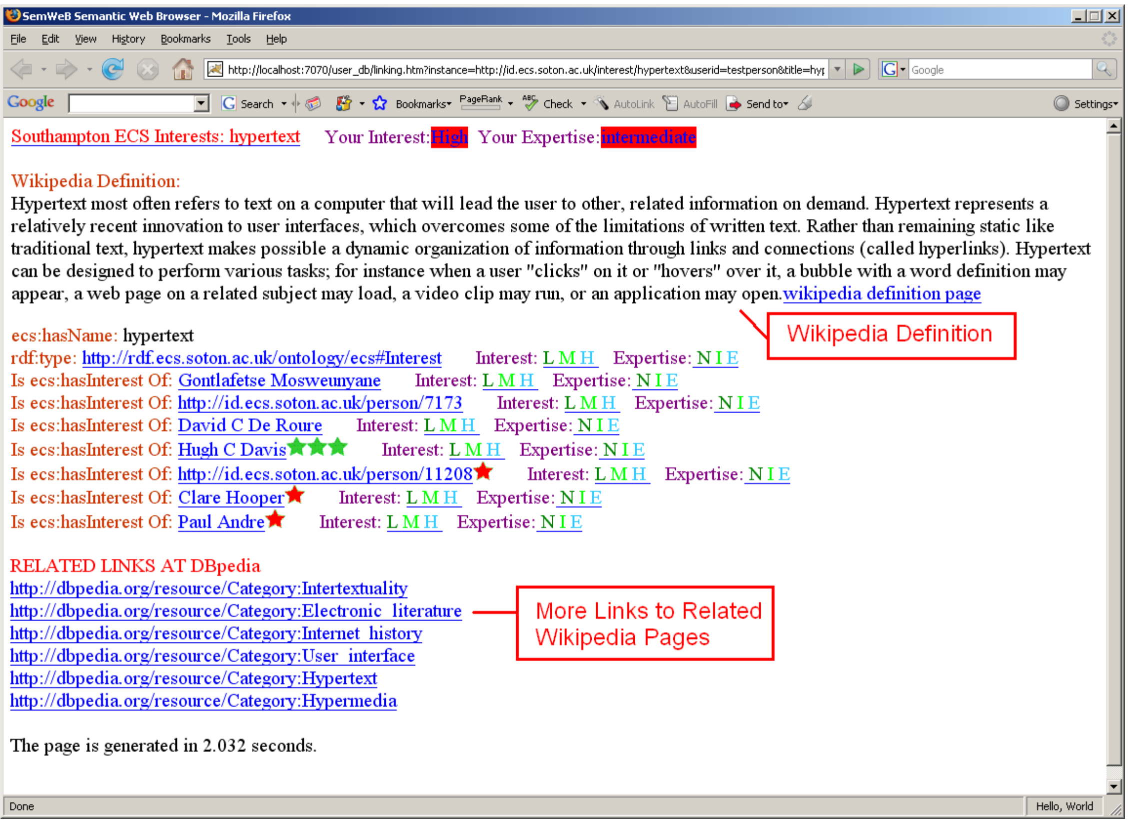
Users can enter interested instances to their profiles together with the ranking value, such as low, medium or high. As shown in Figure 8-14 (b), the user added the *"hypertext"* semantic instance into the profile with rating *"high"*. When the user presses one of ranking values, this makes an AJAX request to the "Add Interest Service" (AIS) which is a Java Servlet hosted within the Web server. Within this request the following parameters are sent: *URI of the interested instance*, *user id* and *interest rating*. The AIS first creates an RDF description for the interest using the user id, URI and interest rating together with the interest creation date. Then, it accesses user profiles database and updates the person's user profile. If the same interest is entered, then information is replaced with the previous one in the database. If a new interest is entered, then this new RDF description is added to the user profile. In addition to updating the user's interests, the browsing behaviour is revised by the AIS and can be changed according to the following conditions:

- If the user's browsing type is *"directed"*, the browsing type remains same.
- If the user's browsing type is *"undirected"* and an interest is added, then browsing type is changed to *"semi-directed"*.
- Otherwise, no change is done on the browsing type.

In the same way, users can indicate their expertise on semantic instances by clicking expertise values, such as novice, intermediate and expert. An AJAX request to the "Add Expertise Service" (AES) is sent together with three parameters: *URI of the instance*, *user id* and *expertise value*. AES is a servlet hosted within Web server. It creates the RDF description for the expertise based on the user id, URI and expertise value. The expertise creation date is also added to this information. Then, the user profiles database is updated with this metadata. If an already added expertise is entered again, then the new information is replaced with the previous one in the database. In Figure 8-14 (a), the user adds an expertise value *"intermediate"* to the semantic instance *"hypertext"*. After the update, the Wikipedia definition and more links to Wikipedia are provided automatically as depicted in Figure 8-14 (b), since novices and intermediates are supplied with explanations and links from DBpedia. This will be explained in more detail later.



(a) Adding interest and expertise values to the user profile



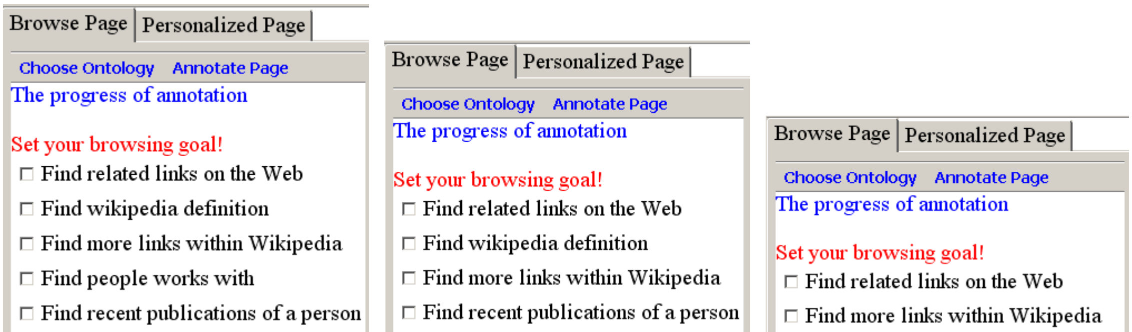
(b) Shows the same page after the addition of interest and expertise values

Figure 8-14 Visualization of interest and expertise addition buttons

8.6.3 Adding Browsing Goals

Browsing goals are automatically provided by SemWeB based on the ontology. All goals are created purposefully to provide useful information to the users, where the information can be obtained implicitly using reasoning, or searched over the Web. Hence browsing goals present indirect knowledge which is not presented within the RDF description of a resource and they may provide useful information to the user.

For displaying available browsing goals, we adopted an ad-hoc solution. In the SemWeB sidebar extension code, all available goal services to each ontology are stored. When the user chooses an ontology from the navigation tab, all accessible browsing goal services are presented with check boxes in the navigation tab. For example, the list of available services to the ECS, DBLP and DBPedia domains are presented in Figure 8-15. To activate these goals, users need to select them. Once the user selects goal(s), the selected services are available for all instances recognized on the Web page. However, the browsing goal is added to the user profile when the user requests semantic information about a semantic instance from the SLS by clicking the explore icons. When the user does that, the SLS checks if the user has any goals for the requested semantic instance. If goals are requested, the SLS first updates the browsing type of the user, where it is set to *"directed"*, and then the corresponding goal services are invoked. All browsing goals are generated as standalone services and can be accessed by the SLS. The details of these services are presented in detail in section 8.8.



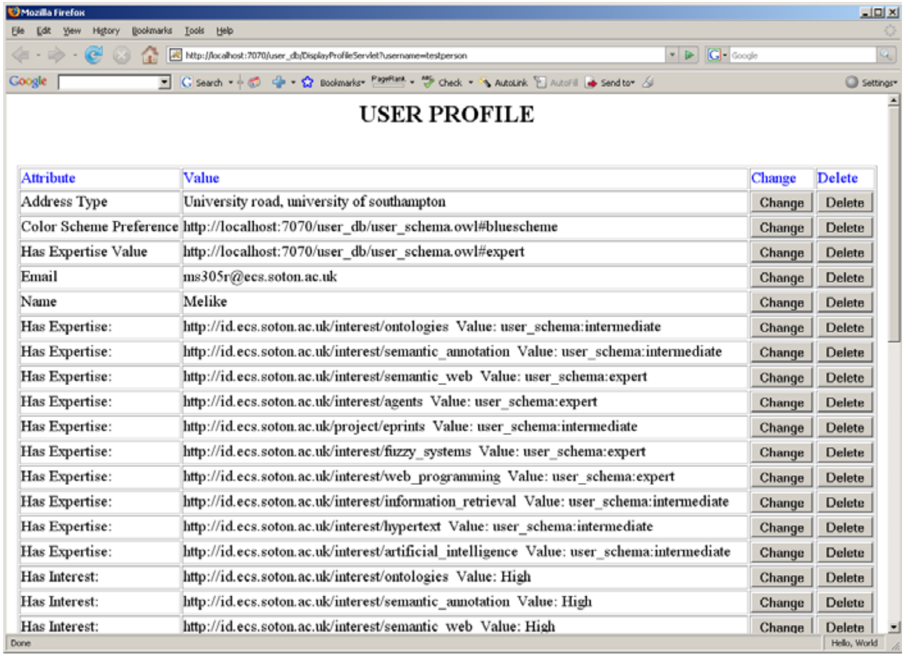
(a) The ECS Domain (b) The DBLP Domain (c) The DBpedia Domain

Figure 8-15 Browsing goals provided by SemWeB according to different ontologies

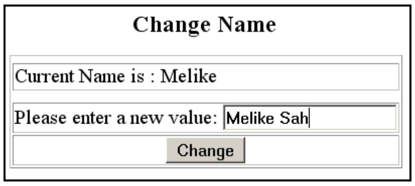
8.6.4 User Profile Updater

Users are often changing their interests and their expertise evolves over time. In order for users to see their profiles and keep them up-to-date, we have provided a profile updater. The profile updater can be accessed from the personalization tab of the SemWeB sidebar using "open/edit profile" menu item. Profile editor is a service, which is a servlet with the Web server. It provides three functions: viewing, editing and deletion.

All of the contents of the user profile are presented when the profile updater is opened (see Figure 8-16 (a)). When the user wants to delete or change existing information, then the Attribute Delete Service (ADS) or Attribute Change Service (ACS) is invoked. Both services take user id, property name and attribute values, then perform searches over the user profiles database for deletion or change. Then, the necessary changes are done and saved permanently to the database. An example is shown in Figure 8-16 (b).



(a) Profile updater interface



(b) Visualization of changes on literal values

Figure 8-16. Profile updater interface provided by SemWeB

8.6.5 Implicit Information Updated by User Heuristic Rules

The browsing interest, browsing type and browsing level concepts are implicitly updated by SemWeB server depending on the actions of the user.

Browsing type: In 8.6.2 and 8.6.3 we have discussed how the browsing type is changed implicitly when the user adds browsing interests or browsing goals. In addition, we use some heuristic rules to update the user profile implicitly.

Browsing interests: These can be understood from the added expertise concepts. For instance, when the user adds an expertise value to a Web resource, then we assume that the user also has an interest in this resource. If the user did not enter an interest value to this resource before, then SemWeB adds this resource to the browsing interests according to the heuristic rules shown in Figure 8-17.

```
If user x has expertise to y -> then user x has interest to y
Predicted interest value according to the expertise value
   If (expertise value == novice) -> then add interest value = low
   If (expertise value == intermediate) -> then add interest value = medium
   If (expertise value == expert) -> then add interest value = high
```

Figure 8-17 A heuristic rule excerpt for adding browsing interests according to the expertise of the user

The implicit user interests are understood when the user adds an expertise value to a semantic resource. When the AES is invoked, the requested expertise value is first added to the profile. Then the AES checks if the user has already added the particular semantic instance to the browsing interests. If not, then the above heuristic rule is applied and the AIS is invoked from the AES with the following values: when the user has an expertise value of "*expert*" then the "*high*" interest value is added to the resource. If the user has an expertise value of "*intermediate*" for the resource, then the "*medium*" interest value is added. Otherwise, the "*low*" interest value is added to the user profile.

Browsing level: In the user model, we have included the browsing level concept. In future work, this concept could be implicitly updated by SemWeB using the number of clicks the user makes in a browsing session.

8.7 Creating Semantic Information and Hyperlinks from the Linked Data on the Request of the User

The main design goal of SemWeB is to guide users during browsing using Semantic Web technologies and AH methods. Previously, we have explained how we identify semantic instances from the Web page and then how users can embed semantic instances into Web pages. The proposed user model was also described, which uses a behaviour-based ontology for Web-based IR adaptation. In this section, we explain the approach that is used in SemWeB for presenting semantic information and links.

Users are supported with semantic information, links and personalized views by using the created semantic annotations and the proposed user model. Users are first required to click on *explore icons* next to the recognized semantic instances on the annotated

Web page. Once they click, the browser makes an AJAX request to the Semantic Linking Service (SLS). The SLS is deployed as a Java servlet hosted within a servlet engine. The SLS takes four arguments as input from the browser:

- URI of a semantic instance
- Lexicon of the semantic instance
- User id of a user (if the user is logged in)
- Goal(s) of the user (if the user has goal(s))

Then, the SLS performs the following steps for the creation of semantic information and links: dereferencing the URI of the resource, creating semantic links and information from the RDF description of the resource, invoking goal services if there are any, invoking the adaptation engine if the user is logged in and converting the response to XML. Since the communication between the browser and the SLS is based on AJAX (see Figure 8-18), responses of the SLS are converted to XML.

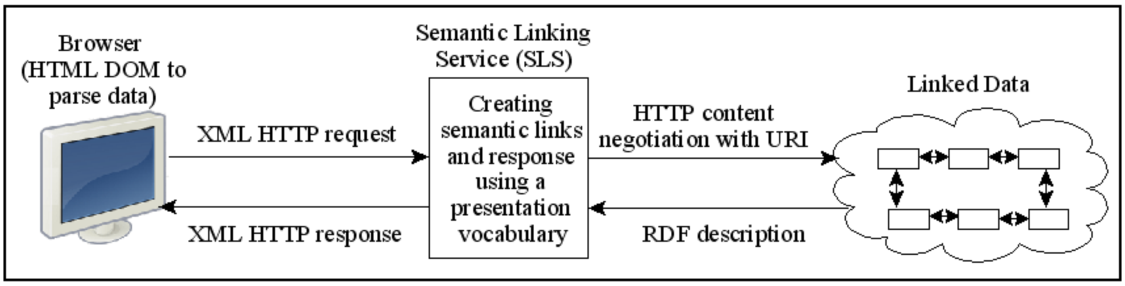


Figure 8-18 Communication between browser-SLS-linked data

8.7.1 Dereferencing a URI

As a first step, the requested URI, assume it is *x*, is dereferenced. If *x* is an HTTP URI, it is dereferenced over the HTTP protocol using content negotiation. If *x* is a # URI, then the # part is stripped off, then dereferenced using content negotiation. Otherwise the URI itself is dereferenced using content negotiation. The RDF description about the resource is cached to a Jena model. The Java code used to perform content negotiation using Jena and HTTP protocol is shown in Figure A-13 in Appendix A. If the RDF description of URI *x* cannot be received successfully, an error message is sent to the browser and the SLS exits. Otherwise, we start to process the RDF description for finding related links and information.

8.7.2 Creating Information and Links from the RDF Description of the Resource

After the RDF description of the URI is successfully received, the following steps are taken to display semantic information and links to the user at the browser.

8.7.2.1 Algorithm for Creating Semantic Links and Information

We use the following algorithm for creating ontology-based links and information about a resource *x* (i.e. dereferenceable URI *x*) as shown in Figure 8-19. After the resource *x* is dereferenced, the literal information (step 2 in Figure 8-19), RDF links (step 3 in Figure 8-19) and inverse RDF links (step 4 in Figure 8-19) are searched using SPARQL queries. We have given more importance to rdfs:seeAlso links, since in the recommended practice of linked data, related resources on the Semantic Web are interconnected using this property. Thus, we present these links separately (step 5 in Figure 8-19). Finally, owl:sameAs links are found using SPARQL queries and added to the response as well. In future work, we could find all owl:sameAs objects using transitive recursion; for example, if (x, owl:sameAs, y) and (y, owl:sameAs, z), then both y and z will be added to the SLS XML response. We are creating XML responses because AJAX allows us to share XML files between the browser and the server. In addition, to view hyperlinks with user-friendly anchor names, we are searching rdfs:label and dc:title properties of a URI. Alternatively, the mapping database is used to find user friendly anchor names to the URIs: the database is queried with a URI and its lexicon is utilized as a link anchor while presenting links to users. If a human-friendly anchor name cannot be found, then we use the URI of the instance.

```
1. Dereference URI x and add graph to a Jena RDF Model.

2. Look up any literal where the graph contains (x, any, y) and (y, isa, literal).
Add results to response XML file using <information> tag.

3. Match triples (x, any, y) and (y, isa, resource). Add retrieved resources into
response XML file using <RDFLinks> tag.

4. Match triples (any, any, x). Add retrieved resources into XML file using
<RDFLinks> tag.

5. Match triples (x, rdfs:seeAlso, y). Add resultant resources into response XML file
with <SeeAlso> tag.

6. Look up triples (x, owl:SameAs, y). Add resultant resources into response XML file
with <SameAs> tag.
```

Figure 8-19 Algorithm for creating semantic information and hyperlinks

Unlike the linked data browsers we discussed in chapter five, we do not provide recursive browsing of linked data. Instead, links to related resources on the Web are supplied using linked data and are personalized according to the user model.

8.7.2.2 Creating Personalized Information and Links

If the user is logged in to the browser, then the created links are personalized according to the user model. This section will be discussed in sections 8.8, 8.9, 8.10 and 8.11. All created adaptive contents are added to the SLS response.

8.7.2.3 Presentation Vocabulary

For sharing the created semantic information between the browser and the server, as well as presenting them at the browser, we have created a simple vocabulary, which we call the *presentation vocabulary*. The tags from this vocabulary are used by the SLS while creating the response in XML. The vocabulary is shown in Figure 8-20.

The PageTitle tag is utilized to present semantic instances with human-friendly names, where the value element contains the user-friendly name of the instance and the target element contains the URI of the instance. The information tag is used for literal value properties; the value element represents the literal value of the property and the property element contains the property name. RDFLink tags are used for RDF links and inverse RDF links between the requested instance and other instances; it contains the value element (user-friendly link anchor name), target element (target URI), property element (relationship name) and similarity element (represents similarity of the URI to the user profile). The SeeAlso tag is used for rdfs:seeAlso links and has value and target elements. The SameAs tag is used for owl:sameAs links and it has value and target elements. WorksWith, WorksWithFrequency, DBPediaLinks, DBPediaAbstract, DBLPRecentPublications and RelatedLinksonWeb are tags used by the SemWeB goal services and are described later.

The SLS uses this vocabulary while creating information about a requested semantic instance and it can be illustrated as follows: assume the user requested semantic information about URI, http://id.ecs.soton.ac.uk/person/1650, from her browser (e.g. the user does not have a browsing goal and is not logged in to the browser – no personalization). Then, the SLS dereferences the URI, applies the algorithm explained in Figure 8-19 and the response in XML is created as shown in Figure 8-21 according to the vocabulary described in Figure 8-20.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:complexType name="PageTitle">
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Information">
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="property" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RDFLink" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
    <xsd:element name="property" type="xsd:string"/>
    <xsd:element name="similarity" type="xsd:double"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SeeAlso" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SameAs" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="WorksWith" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
    <xsd:element name="frequency" type="xsd:double"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="WorksWithFrequency" >
  <xsd:sequence>
    <xsd:element name="maxfrequency" type="xsd:integer"/>
    <xsd:element name="minfrequency" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DBPediaLinks" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DBPediaAbstract" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DBLPRecentPublications" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
    <xsd:element name="target" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RelatedLinksonWeb" >
  <xsd:sequence>
    <xsd:element name="value" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 8-20 XML schema presentation of the presentation vocabulary

```xml
<?xml version="1.0" ?>
<message>
      <PageTitle>
        <value>Southampton ECS People: Melike Sah</value>
        <target>http://id.ecs.soton.ac.uk/person/9677</target>
      </PageTitle>
      <Information>
        <value>Melike Sah</value>
        <property>hasFullName</property>
      </Information>
      <RDFLinks>
        <value>semantic web</value>
        <target>http://id.ecs.soton.ac.uk/interest/semantic_web</target>
        <property>hasInterest</property>
        <similarity>0</similarity>
      </RDFLinks>
      <RDFLinks>
        <value>Designing a Personalized Semantic Web Browser</value>
        <target>http://id.ecs.soton.ac.uk/presentation/290</target>
        <property>Is hasPresenter Of</property>
        <similarity>0</similarity>
      </RDFLinks>
      ...
</message>
```


Figure 8-21 An XML response created by SLS (no adaptation)

It is also worth mentioning that the Fresnel vocabulary can be used to present information at the browser (Pietriga et al., 2006). Fresnel is a browser-independent presentation vocabulary for RDF. It relies on two concepts: lenses and formats. Lenses are used to specify which properties to show and their presentation order, while the format indicates the style of the presentation using Cascading Style Sheets (CSS). In this thesis, in addition to the RDF presentation, we are presenting information and links from goal services and personalization. Hence, we created a simple vocabulary, which involves that information. However, for the RDF presentation, the Frensel vocabulary can be utilized later.

8.7.2.4 Displaying Created Semantic Information and Links at the Browser

The final step is the presentation of the created information at the browser. The data is sent back to the browser in XML and received using XMLHttpRequest. The HTML DOM object (W3C, 2003) is utilized to parse XML data and present information and links in a new Web page using CSS as shown in Figure 8-22. In this step, we use CSS to display similar information with same format.

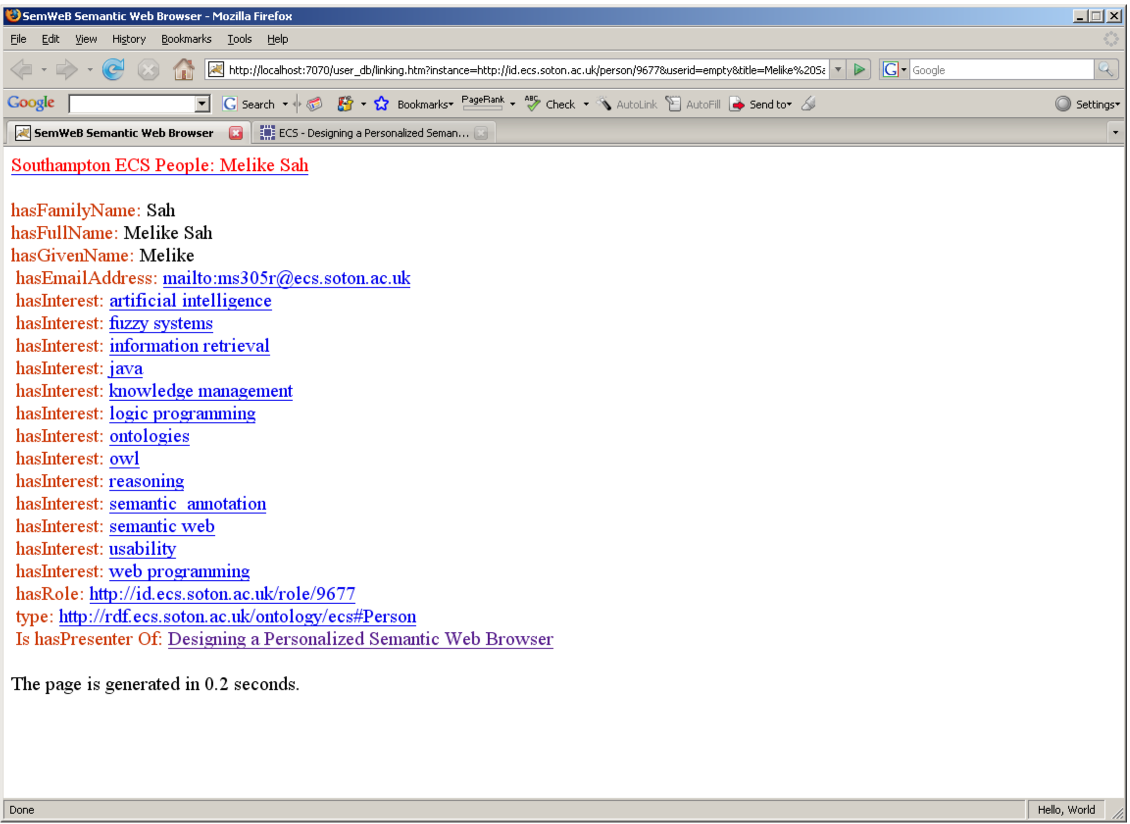


Figure 8-22 An example of semantic information and links presented at the browser without adaptation

In the next sections, we explain how the semantic links created by the SLS is adapted to individual users using the proposed behaviour-based user model. We have utilized different AH techniques and methods to achieve this, such as adaptive link generation, link annotation with visual cues and adaptive text presentation.

8.8 Adaptive Goal-Based Link Generation

In SemWeB, adaptive links are generated according to the browsing goals of the user. In the proposed behaviour-based user model, the user's browsing needs are represented by the browsing type concept. If the user's browsing type is *"directed"* (which means the user is looking for a specific information), then adaptive links are generated according to the browsing goal(s). In our opinion, it could be more convenient for users, if goals are supplied proactively. Thus, different goals are automatically provided depending on the selected ontology domain. In future work, user-based manual goals could be supported by using a goal search-box from the navigation tab.

SemWeB utilizes goal services to present dynamic and adaptive links according to the current information needs of the user. Two different approaches are used by the goal services while creating adaptive links: either implicit information about the instance is shown (i.e. additional information is presented which does not exist in the RDF description of the instance) or related semantic information is looked for on the Web (i.e. searches on other linked data sets are performed).

Depending on the selected goal, the requested instance and the ontology, each service formulates a dynamic SPARQL query. Then, the query is either executed on the RDF description of the instance or other datasets available on the Semantic Web. Finally, the resultant links are parsed and added to the SLS XML response. In this way, while displaying semantic links and information to the users, additional links to other related resources are presented according to the user's desired browsing goals.

Goal services can be categorized in two groups: general goals and domain-specific goals. As its name infers, general goals are application-independent and can be used for diverse instances on different domains. For instance, the "DBpedia Definitions" goal service can be applied to different class instances of diverse ontologies. On the other hand, domain-specific goals relate specifically to certain class instances of an ontology domain. An example is the "People Works With" goal service in ECS. In the rest of this section, the goal services used in the ECS domain are explained in detail.

8.8.1 Searching for Related Semantic Information on the Web (SRSIW) Goal Service

One of the services provided by SemWeB is "*Searching for Related Semantic Information on the Web (SRSIW)*". SRSIW is a general goal service and applicable to different ontological instances of any ontology. It is mainly employed to find related information about a semantic instance on the Web. This is done by utilizing Sindice[54]. Sindice is a Semantic Web search engine, which takes URIs or keywords as inputs, then queries millions of indexed Semantic Web content and returns results.

Sindice has two types of search formats: term search and advanced search. Term search lets you retrieve documents related to given keywords or URIs (shown in Figure 8-23

[54] http://sindice.com [last accessed, 1/1/2009]

(a) and 8-23 (b)). In the advanced search, triple queries can be submitted to the Sindice. For instance, in Figure 8-23 (c), we are searching RDF files that contain foaf:name "Wendy Hall". The results from Sindice can be retrieved in three different formats by negotiating the content: in json, rdf+xml or xml.

```
http://api.sindice.com/v2/search?q=%22Cyprus%22&qt=term
```

(a) The Sindice keyword term search

```
http://api.sindice.com/v2/search?q=http://id.ecs.soton.ac.uk/person/9677&qt=term
```

(b) The Sindice URI term search

```
http://api.sindice.com/v2/search?q=*+<http://xmlns.com/foaf/0.1/name>+%22Wendy+Ha
ll%22&qt=advanced
```

(c) The Sindice advanced search

Figure 8-23 Examples of Sindice term search, URI search and advanced search

SRSIW for dynamic querying the Semantic Web using Sindice: Users can activate SRSIW, by checking "*find related links on the Web*" checkbox from the navigation tab of the SemWeB sidebar. When the user requests semantic information about a semantic instance by clicking the explore icon, this service request is also passed to the SLS together with the instance URI and its lexicon. The SLS then invokes SRSIW to find related links about this particular instance. SRSIW uses keyword term search such that it creates a search request to Sindice using the lexicon of the instance as shown in Figure 8-23 (a). This request is sent to the Sindice server using the HTTP protocol and the content type is set to rdf+xml. If Sindice is accessed over HTTP, a maximum of 10 results are sent back, such that it contains all RDF files that mention the requested keyword. The results are sent back to SRSIW in RDF and cached to a Jena Model (as shown in Figure 8-24 (a)). We extract links from this RDF model using SPARQL queries and links are added to the SLS response using "RelatedLinksonWeb" tag as shown in Figure 8-24 (b). In this way, semantic links to other resources on the Web can be found and presented during the presentation of semantic links as presented in Figure 8-25. In addition to keyword term search, the URI of the instance is also searched using the Sindince URI term search and the links found are added to the SLS response. The SRSIW service is ontology-independent and can be used on different domains.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://sindice.com/vocab/search#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
  <Result rdf:about="http://api.sindice.com/v2/search?q=%22photography%22#result1">
    <rank>1</rank>
    <content>96 triples in 14388 bytes</content>
    <fields:format>RDF</fields:format>
    <dc:created>2008-12-08T00:00:00+00:00</dc:created>
    <link rdf:resource="http://dbpedia.org/resource/Category:Photography"/>
    <dc:title>Photography</dc:title>
  </Result>
```

(a) An example rdf+xml response returned by the Sindince

```
<RelatedLinksonWeb>
<value>http://dbpedia.org/resource/Category:Photography_equipment</value>
</RelatedLinksonWeb>
```

(b) Links added by SRSIW according to the presentation vocabulary

Figure 8-24. (a) Sindice rdf+xml response (b) Added SRSIW links by SemWeB

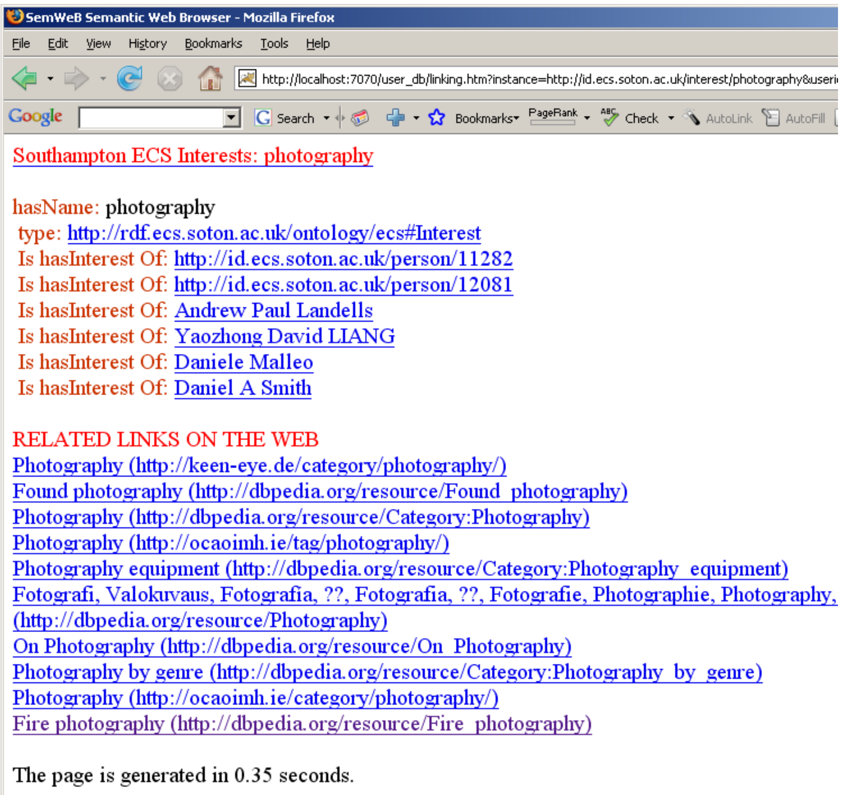


Figure 8-25 Demonstration of the created adaptive links by SRSIW goal service

8.8.2 People Works With (PWW) Goal Service

On the ECS domain, many of people listed are staff and postdoctoral students. In their RDF files, research related information is presented, such as RDF links to interests, publications and projects that person involved. During browsing, if a user is interested in a person, then she might be interested in which people work on the same or similar projects. Therefore, to provide implicit information that is not presented in the person's RDF file, but might be useful to the reader, we supplied "*People Works With (PWW)*" service. PWW is used to show related people that work with a person based on shared projects. Hence, this service is available for ECS Person instances and is an application-specific goal service.

PWW can be activated by selecting the "*Find people work with*" checkbox from the navigation tab of SemWeB sidebar. When user request information about a person and this service is selected, the PWW hosted within the SLS starts to search for people who work on the same projects. Figure 8-26 illustrates the working mechanism of the PWW.

```
Query projects that a person x is involved
 For each project, dereference project URI to a temporary Jena model
  If (project_end_date > today)
    Find people works on the project using SPARQL queries
        For each person y and y≠x, increase the counter for shared projects and add
        person y's URI and counter into a vector
         End
    End
  End
 End
```

Figure 8-26 Algorithm for finding people who work on the same projects

PWW first searches for the projects that the requested person is involved in. In ECS, previous and current projects are listed in a person's RDF file. To eliminate previous projects, first we dereference the project URI and filter the end date of the project. If the project is still continuing, then other members of the project are queried. For each person URI, we keep a record of shared projects with the requested person. For instance, person *y* works on three projects together with the requested person. Finally, using the presentation vocabulary of SemWeB, people who work with the requested person are added to the XML response of the SLS. For instance, the following XML tags are created after running the PWW and added to the SLS response.

```
<WorksWith>
  <value>Prof Nicholas Jennings</value>
  <target>http://id.ecs.soton.ac.uk/person/2355</target>
  <frequency>3</frequency>
</WorksWith>
<WorkswithFrequency>
  <maxfrequency>3</maxfrequency>
  <minfrequency>1</minfrequency>
</WorkswithFrequency>
```

Figure 8-27 Created XML response to the browser after running the PWW service

To prioritise people who work on more shared projects with the requested person, we present this information as a tag cloud at the browser (as shown in Figure 8-28). Minimum frequency and maximum frequency are used to create the tag cloud. We use different CSS classes to set the size of the link anchors using the tag cloud font distribution algorithm[55].

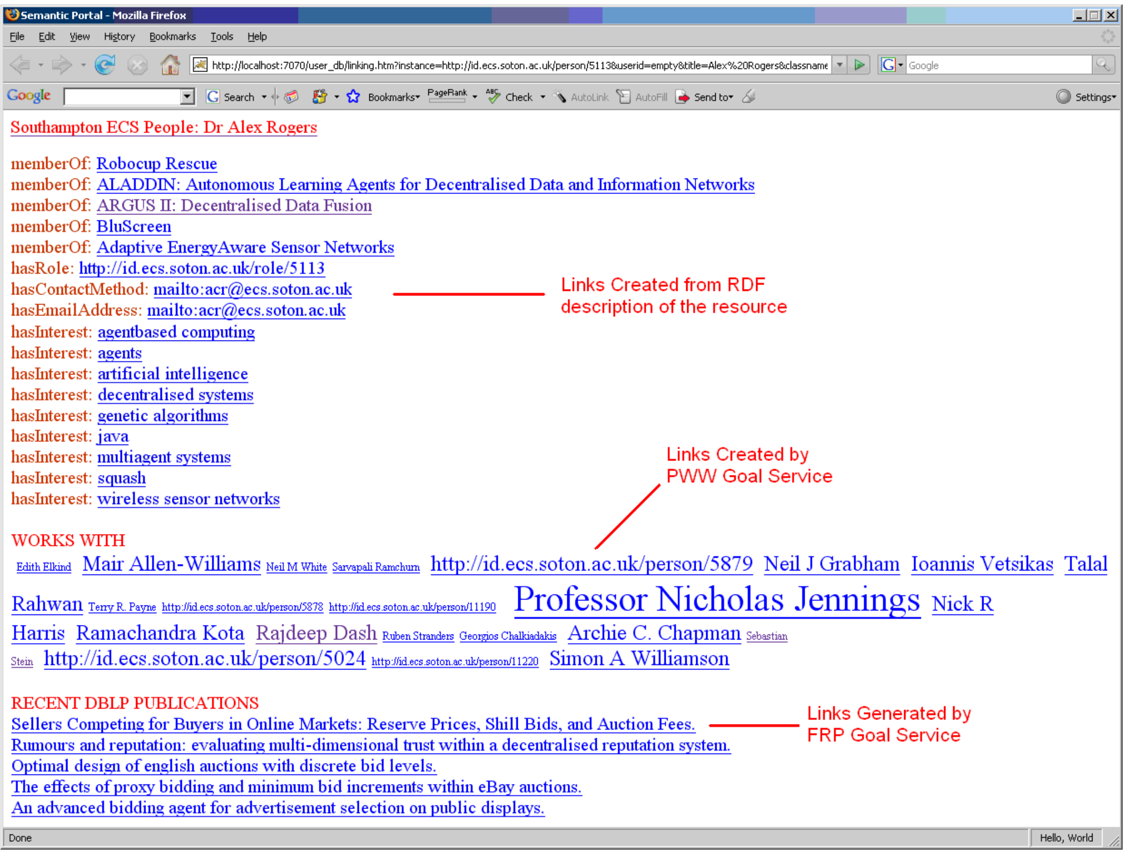


Figure 8-28 Demonstrations of the created adaptive links from PWW goal service and FRP goal service

8.8.3 *Find Recent Publications of a Person (FRP) Goal Service*

In the ECS domain, all person RDF files contain information about publications. However, most of the profiles have an incomplete set of publications. In a research environment like ECS or in other education-related ontologies, the recent publications of a person could be useful information. To present the most recent publications, we use DBLP (DBLP, 2008). The DBLP database is updated every week and contains a

[55] http://www.petefreitag.com/item/396.cfm [last accessed, 3/1/2009]

complete list of publications of a person. It provides a public SPARQL endpoint[56] and can be searched over the HTTP protocol using SPARQL queries. To search over this database, we have developed the "*Find Recent Publications of a Person (FRP)*" goal service. FRP can be used on different person instances of diverse ontologies and it is a general goal service.

FRP is activated when the user selects the *"find recent publications of a person"* goal service from the navigation tab of the SemWeB sidebar. When, the user requests a person instance from SLS, the FRP service is activated and invoked from the SLS. Then the FRP performs the following steps to find recent publications of the requested person as shown in Figure 8-29.

```
Search for DBLP URI of the person (Sindice search using lexicon of the person
instance)
If (a DBLP URI is found, then check DBLP foaf:name with the instance's lexicon)
   If (there is a match)
       Construct a SPARQL query for searching recent publications
       Request the query from DBLP SPARQL endpoint
       Cache results to a Jena model
       Extract links and add to SLS XML response
   End
End
```

Figure 8-29 The algorithm that is used by FRP goal service

As a first step, the DBLP URI of the person is searched based on the lexicon of the person instance using Sindice. In the DBLP, "foaf:name" property is used to present resources with user-friendly names. The search results are checked one by one and the foaf:name property of the search result URIs is compared with the lexicon. If they are same, we assume that the DBLP URI is correct. The next step is to query the DBLP database for the recent publications of this person. To achieve this, a dynamic SPARQL query is formed based on the DBLP URI of the requested person (see Figure 8-30 (a), where dblp_uri represents the URI found by Sindice). Then, this query is requested from the DBLP SPARQL endpoint. The requested SPARQL query uses the CONSTRUCT form, which means the results can be obtained in rdf/xml format using content negotiation over the HTTP protocol. Finally, the FRP caches the resulting RDF graph into a Jena model and extracts links and inserts them into the SLS client response using the presentation vocabulary (see Figure 8-30 (b)). For instance, the label of the

56 http://dblp.l3s.de/d2r/sparql [last accessed, 9/9/2008]

page is used as a link anchor and the HTML homepage of the publication is used as the target of the link. Examples of links generated by FRP are shown in Figure 8-28.

```
CONSTRUCT
{<"+dblp_uri+"> <http://xmlns.com/foaf/0.1/made> ?page.
?page <http://www.w3.org/2000/01/rdf-schema#label> ?label.}
WHERE
{ ?paper <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Document>.
?paper <http://www.w3.org/2000/01/rdf-schema#label> ?label.
?paper <http://xmlns.com/foaf/0.1/homepage> ?page.
?paper <http://purl.org/dc/elements/1.1/creator> <"+dblp_uri+">.
?paper <http://purl.org/dc/terms/issued> ?year.
FILTER regex (str(?year), \"2008\")}";
```

(a) SPARQL query for finding recent publications of a person from DBLP

```
<DBLPRecentPublications>
   <target>http://dx.doi.org/10.1145/1329125.1329326</target>
   <value>Rumours and reputation: evaluating multi-dimensional trust within a decentralised reputation system.</value>
</DBLPrecentPublications>
```

(b) Links added by FRP according to the presentation vocabulary

Figure 8-30 SPARQL query used to search DBLP and the created SLS response

8.8.4 DBpedia Definitions (DBD) Goal Service

DBPedia is an extensive public database, which provides semantic information about many world objects (i.e. people, places, books, scientific research areas, artists, albums, movies, etc.). When a user needs to understand a concept on a Web page, DBPedia pages can be used to offer explanations, definitions and more links about the concept. DBPedia supports a public SPARQL endpoint[57], which can be accessed over the HTTP protocol. The DBPedia database can be queried in detailed SPARQL queries using the DBPedia vocabulary from this public endpoint. Thus, to provide links to DBPedia definitions, we have implemented the *"DBpedia Definitions (DBD)"* goal service. The service reformulates detailed SPARQL queries and then accesses and searches the DBPedia database over the HTTP protocol. DBD is a general goal service and applicable to different ontology instances.

DBD is activated when the user selects the *"find Wikipedia Definition"* goal service from the navigation tab of the SemWeB sidebar extension. When, the user requests a

[57] http://dbpedia.org/sparql, [last accessed, 9/9/2008]

semantic instance from the SLS, the DBD service is accessed by the SLS. The DBD service simply creates a dynamic SPARQL query depending on the requested instance (see Figure 8-31 (a)). The lexicon of the instance is used to query DBpedia, for instance, DBpedia resources that contain this lexicon in the "rdfs:label" property are searched. In the DBpedia vocabulary, resource definitions are represented by "http://dbpedia.org/property/abstract" property. The aim of the DBD goal service is to find the value of this property. Since, DBpedia supports diverse language representations for DBpedia abstracts, in the SPARQL query a language filter is used. By default, the English version of the DBpedia definition is retrieved. In addition, the DBpedia definition can be personalized by using the preferred language of the user from the user model.

```
CONSTRUCT
{?x <http://dbpedia.org/property/abstract> ?abstract. }
WHERE
{?x <http://www.w3.org/2000/01/rdf-schema#label> "+lexicon+"@en.
?x <http://dbpedia.org/property/abstract> ?abstract.
FILTER (lang(?abstract)= "en")}
```

(a) SPARQL Query to find the DBpedia definition of a resource

```
<DBpediaAbstract>
 <value> Usability is a term used to denote the ease with which people can employ a
 particular tool or other human-made object in order to achieve a particular goal.
 ...
 </value>
 <target>http://dbpedia.org/resource/Usability</target>
</DBpediaAbstract>
```

(b) DBPedia definition added by the DBD using the presentation vocabulary

Figure 8-31 SPARQL query used to find DBpedia definitions and the SLS response

By using the SPARQL query shown in Figure 8-31 (a), the DBPedia SPARQL endpoint is queried over the HTTP protocol and the results are cached to a Jena model. Then, the definition of the resource is extracted and added to the SLS XML response using the presentation vocabulary as shown in Figure 8-31 (b). In Figure 8-33, the added DBpedia definition is shown at the browser.

8.8.5 Find More Links within DBPedia (FMLDB)" Goal Service

In the previous goal service, we showed that DBPedia definitions can be used to help users to understand a concept. In addition, DBPedia provides links to related Web resources using the SKOS vocabulary. For example, DBPedia provides links to broader topics and links to subjects which contain a specific concept. In our opinion, these links can be utilized to show users more related Web resources within DBPedia. To achieve this, we have implemented the *"Find More Links within DBPedia (FMLDB)"* goal service. The service uses dynamic SPARQL queries to search over the DBPedia database using the HTTP protocol. FMLDB is a general goal service and applicable to different domain ontology instances.

```
CONSTRUCT
{?x <http://www.w3.org/2004/02/skos/core#related> ?subject.}
WHERE {?x <http://www.w3.org/2000/01/rdf-schema#label> "+keywordtitle+"@en.
{?x <http://www.w3.org/2004/02/skos/core#subject> ?subject.}
UNION
{?x <http://www.w3.org/2004/02/skos/core#broader> ?subject.}
UNION
{?subject <http://www.w3.org/2004/02/skos/core#broader> ?x.}}
```

(a) SPARQL query to find more related links within DBpedia

```
<DBpediaLink>
  <value>http://dbpedia.org/resource/Category:Ergonomics</value>
  <target>http://dbpedia.org/resource/Category:Ergonomics</target>
</DBpediaLink>
```

(b) DBPedia Links generated by FML goal service using the presentation vocabulary

Figure 8-32 SPARQL query to search DBpedia links and the created SLS response

FMLDB is activated when the user selects the *"Find more links within Wikipedia"* goal service from the navigation tab of the SemWeB sidebar extension. When the user requests a semantic instance from the SLS, the FML service is invoked by the SLS together with the lexicon of the semantic instance. FMLDB formulates a dynamic SPARQL query for searching DBpedia resources that have this lexicon in the "rdfs:label" property. Using the SKOS vocabulary, "skos:broader" and "skos:subject" properties of the found DBpedia resources are searched. The created SPARQL query that is shown in Figure 8-32 (a) is then executed over the HTTP protocol from the DBPedia SPARQL endpoint. In the query, we are searching for three graph patterns, thus we have used UNION to merge those graphs. Then, results are stored to a Jena model. The links from the model are then extracted and added to the SLS response using the presentation vocabulary (see Figure 8-32(b)). Finally, the created links are presented to the user at the browser as shown in Figure 8-33.

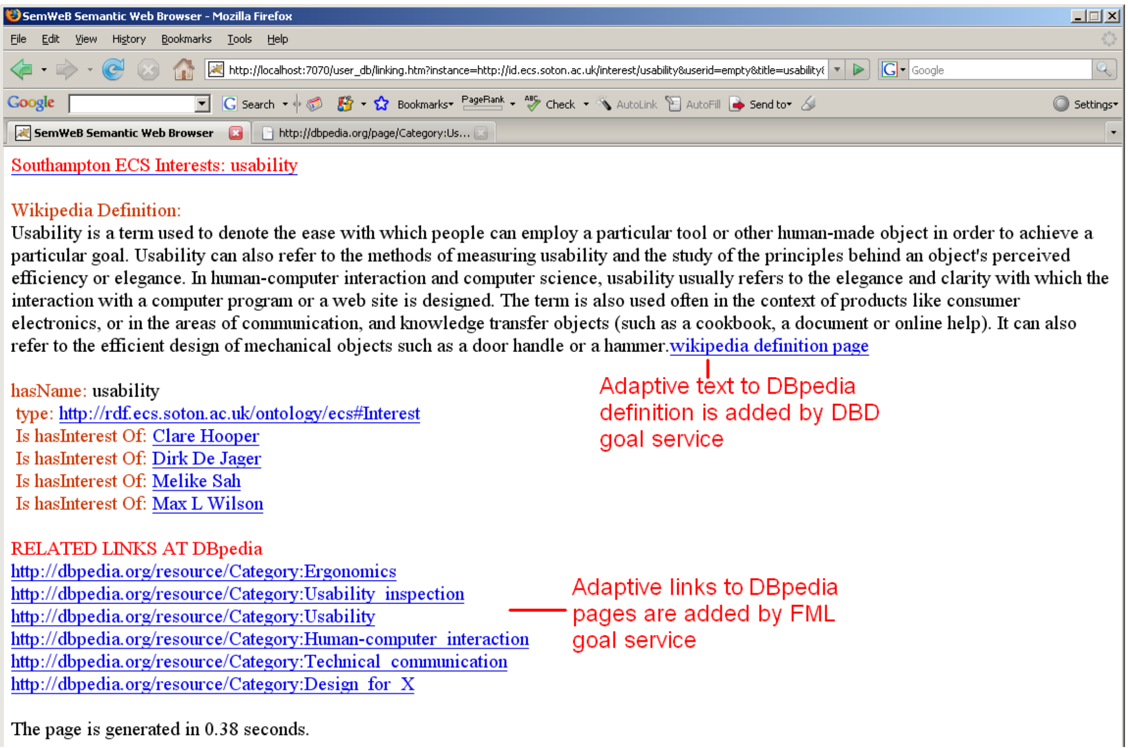


Figure 8-33 DBPedia definition and DBPedia links are presented at the browser

8.9 Link Annotation with Visual Cues Derived from Calculating Semantic Relatedness between the User Model and the Semantic Hyperlinks

In the previous section, we explained that when the user's browsing type is directed, adaptive links are created according to the browsing goals (i.e. links to DBPedia definitions or recent publications, etc.). When the user's browsing type is semi-directed (i.e. the user has browsing interests in a browsing session) or undirected (i.e. no browsing goal and no browsing interest added by the user), then the current and previous interests of the user can be used to personalize semantic hyperlinks created by SLS. For instance, each semantic link that is created by the SLS has a dereferenceable URI. The RDF description of the semantic link can be obtained by using content negotiation. In addition, all user interests in the user profiles are pointing to the dereferenceable URIs of the semantic instances. In the same way, the RDF description of the user's interests can be reached again using content negotiation. As a result, the RDF description of the user's interests and the RDF description of the semantic links are available at the time of creating links. By using these semantic metadata, the semantic relatedness of the links to the user profile can be calculated. Thus, semantic links can be presented adaptively with different visual cues based on the similarity measures between the user's interests and the link's metadata.

One of the biggest advantages of using the Semantic Web standards lies here. Since all links and user interests are machine-processable, it is possible to achieve AH on different domains. The Semantic Web provides an opportunity to accomplish open-corpus AH and our aim is to achieve this within SemWeB. The user's browsing experience can be seamlessly personalized without users having to know about the details of the Semantic Web.

SemWeB uses ontology-based metadata and we need to utilize an ontology-based similarity measure in our system. In addition, there are other criteria that we want to achieve. For example, SemWeB should be able to work on different domains. In SemWeB this is achieved with a scalable and interoperable system design: we adopted Semantic Web technologies for representing metadata about Web resources and user profiles. In addition, linked data is utilized, thus metadata is not restricted to one domain and SemWeB can be adapted to different ontologies with small changes in the IE and semantic annotation. Furthermore, to accomplish open-corpus AH, the similarity measure should be ontology-independent.

An investigation of ontology-based similarity measures was performed. From this literature review, we have seen that similarity measures might be ontology dependent and different techniques may be suitable for different ontologies. Before, explaining the proposed similarity measure, we would like to explain the basic concepts of ontology-based similarity measures. In the rest of this section, first the difference between semantic similarity, semantic distance and semantic relatedness is given. Generic aspects of similarity measures are briefly discussed. Then, research in ontology-based semantic similarity measurement is summarized. Finally the proposed user-based semantic relatedness measure is given.

8.9.1 Semantic Similarity/Distance/Relatedness Measures

Determining the semantic similarity between words has been widely used in the history of philosophy, psychology and artificial intelligence. Semantic similarity is a concept where a set of documents or lists of structured terms is assigned to a metric according to likeliness of their semantic content or meaning. The roots of semantic similarity measuring is from natural language processing applications, such as information extraction, retrieval, text summarization, automatic indexing, etc. The three different terms *similarity*, *distance* and *relatedness* are sometimes used interchangeably in the research literature. These terms however are not the same.

Semantic relatedness measures include a variety of relationships, such as meronymy (car-wheel), synonymy, functional (car-gasoline), associative (car-speeding ticket), and subsumption in calculating the value. In other words, the semantic relatedness makes use of combination of different relationships between concepts. On the other hand, semantic similarity uses only synonymy and the subsumption relations in the calculation. The semantic distance measures the distance between concepts within a network of structure (e.g. ontologies). In the research literature, the semantic distance is accepted as the antonym term to the semantic similarity and the semantic relatedness, which is the view accepted in this thesis.

Semantic similarity and semantic relatedness measures can be categorized based on three underlying approaches: distance-based within a network structure, information content-based and hybrid approaches. In distance-based methods (also known as edge-based) the number of edges between nodes is calculated. Information-content based techniques measure the similarity between the content of each node and computationally complex than edge-based methods since they need to compare each node. Hybrid techniques combine both edge-based and node-based methods.

In this thesis, our main focus is to find an appropriate similarity measure for use with ontologies. Thus, we investigated the semantic similarity measures used in ontologies.

8.9.2 Semantic Similarity Measurements in Ontologies

According to (Bernstein et al., 2005), there are two different ways which use hierarchical ontology structure for determining the semantic similarity between objects in an ontology: the edge-based approach and the node-based approach. Edge-based techniques calculate the distance/edge length between nodes in an ontology. The longer the path, the less similar they are. In this approach, the problem is that all edges have equal weight and performance depending on the construction of the ontology. On the other hand, node-based techniques measure content-based similarity, for instance shared information and relationships between concepts are predicted.

In the literature on ontology-based similarity measures, Bernstein et al. 2005 summarizes five different distance measures, both node and edge based: ontology distance, information-theoretic approaches, vector space approaches, edit distance and full-text retrieval method ($tf \times idf$). The semantic distance is used as the antonym term to the semantic similarity and the semantic relatedness, the higher the distance, the less similar the given two concepts.

Ontology Distance: This measuresthis measure the distance between objects in an ontology. The calculation of the ontology distance is according to is-a relationship specification of graph of objects (Rada et al., 1989; Lee et al., 1993). The ontology is a directed acyclic graph of specialized objects. Examples are WordNet[58], thesaurus or other ontologies that represent taxonomies. The ontology distance between two objects is calculated through a common ancestor. The distance is the sum of the length of the path from the first object to a common ancestor and the ancestor to the second object.

Information-theoretic Approaches: The problem with ontology distance is that it is highly dependent on subjective ontology engineering choices. Therefore, the information-theoretic approaches try to overcome this problem by measuring similarity between two objects in an ontology by means of shared information between them (Resnik, 1999; Lin, 1998). There are different methods, for example probabilistic methods, to measure the degree of overlap of descendents of two objects in hierarchical ontology like WordNet.

Vector Space Approaches: In vector space, each object is represented with a vector of features in a k-dimensional space. The similarity is measured by cosine or Euclidean distance of vector features of two objects. In an ontology, the vectors contain attribute values of an object (Castells et al, 2007). But in this type of vectorization semantically closer attributes are not captured.

[58] http://wordnet.princeton.edu/ [last accessed, 3/1/2009]

Edit Distance (Levenshtein Distance): This measures the similarity between strings using the edit distance, the number of changes needed to convert one string into another (Levenshtein, 1966). It contains insertion, removal or replacement of symbols. It counts the number of transformation steps taken to convert one object to another. In an ontology, the Levenshtein distance measures how many steps need to be taken to convert parts of one object (attributes/relationship) to the parts of other object.

Full-text Retrieval Method ($tf \times idf$): This is the most often used method in the information retrieval literature. It compares two documents using "term frequency and inverse document frequency" weighting scheme (abbreviated to $tf \times idf$) (Baeza-Yates and Ribeiro-Neto, 1999). If we simply summarise it, $tf \times idf$ counts the frequency of occurrences of a term in a document and relates it to the frequency of occurrence of that term in the entire corpus of documents. The resulting weights are combined into weighted term vectors describing the document. Then, two documents are compared as the cosine between their weighted term vectors. In an ontology, object attributes and their values can be used to create weighted term vectors and these vectors can be compared for similarity measurement.

8.9.3 Discussion of Semantic Similarity Measures in Ontologies

The ontology distance measure is highly dependent on the ontology structure and applicable to ontologies representing taxonomies or thesauri. In our case, we have used the ECS ontology (ECS Ontology, 2008) for experimentation with SemWeB. The ECS ontology is not a good candidate for the ontology distance based similarity measure since the ontology was not constructed for information retrieval purposes. On the other hand, information-based approaches are complex in computation since they require the comparison of every attribute of an object. Vector space, edit distance and full-text retrieval methods are useful but they do not take into account semantic relationships coming from the ontology structure. In our system, we intend to utilize a computationally-efficient technique, as well as similarity measure that should be proficient and applicable to different ontologies, since we aim to achieve open-corpus AH. Additionally, the preferences of users should be considered during the calculation of similarity measure and in the next sections, first we discuss related work then explain the proposed user-based semantic relatedness measure.

8.9.4 Related Work

In research literature, ontology-based similarity/relatedness measurements are used in clustering (Maedche and Zacharias, 2002), recommendation systems (Ehrig et al., 2004), ontology mapping (Maedche and Staab, 2002) and many other purposes. For instance, (Ehrig et al., 2004) presents a comprehensive framework for comparing similarity between entities within and between ontologies in a peer-to-peer network in Bibster system[59]. In this approach, similarity between entities can be measured based on four criteria, such as data-level similarity (e.g. comparison of string values), ontology-level similarity (e.g. similarity according to distances of objects within the ontology) and context-level similarity (e.g. comparison of entity usage in an external content) and domain knowledge-level similarity (e.g. comparison of domain specific features). Then, all of the individual measurements are amalgated. These measures are utilized to perform duplicate detection, peer selection and recommendations of publications in the Bibster system. (Maedche and Zacharias, 2002) also presents a similarity measure for ontology-based metadata for hierarchical clustering. This approach combines three different dimensions in the calculation, such as taxonomical similarity (the similarity of entities according to their position in the ontology), relation similarity (similarity based on relations to other objects) and attribute similarity (similarity of two entities based on attributes and their values). Then similar concepts are clustered. (Maedche and Staab, 2002) shows a similarity measure for comparing ontologies. For instance, they perform lexical comparison (e.g. Levenshtein edit distance to compare strings and numbers) and taxonomical comparison for super- and sub-concepts comparison.

The researcher observed that different similarity measures are appropriate in different scenarios and use-cases. The similarity or relatedness measure depends on the ontology, ontology-based metadata and the purpose of the application.

8.9.5 The Proposed Semantic Relatedness Measure

Semantic similarity measures are widely used in the literature for calculating relatedness. Most of the methods discussed in this literature use WordNet to

[59] http://bibster.semanticweb.org/ [last accessed, 24/3/2009]

demonstrate the quality of the similarity measure. WordNet is a hand-crafted taxonomy created by Princeton University and especially developed for information retrieval purposes. Hence, it classifies concepts in a well-defined taxonomical hierarchy and subsumption or synonyms based techniques performs well. However, there are many ontologies or network-based structures that are designed for information sharing and do not address information retrieval efficiently. For instance, the ECS and DBLP ontologies are not suitable for ontology distance based semantic similarity measures; relying only on subsumption is not a good similarity measure. Thus, we adopted a semantic relatedness measure, where various relationship edges from the ontology are utilized in the calculation of the similarity value. We weighted different relationship edges differently for the calculation of the similarity measure.

First, we formalized the concepts of similarity, using the definition of a similarity function introduced by (Richter, 1992).

Similarity Measure: A similarity measure is a function $sim(x, y): U^2 \rightarrow [0,1]$ on a set of U measuring the degree of similarity between x and y.

There are split opinions about the properties of sim, but it is generally agreed that

$$\forall x, y \in U \qquad \begin{array}{c} sim(x,x) = 1 \text{ (reflexivity)} \\ sim(x,y) = sim(y,x) \text{ (symmetry)} \end{array} \tag{8-1}$$

In our algorithm, if the two compared objects are same, then the similarity is 1 (reflexivity). In addition, the similarity between x and y equals the similarity between y and x (symmetry).

8.9.5.1 Syntactic similarity

In SemWeB, our aim is to compare similarity between two ontology instances, where instances may belong to two different ontologies. In the case, where the two compared instances are coming from different ontologies we compare the *syntactic similarity* of instance titles. (Levenshtein, 1966) introduced a measure to compare two strings using the edit distance and we use this measure to compare titles of instances. Particularly informative types of literals such as “rdfs:label”, “dc:title” and “foaf:name” are used to define human-readable version of a resource’s name and they are domain-independent. For each instance thus in SemWeB, first we search for human-friendly names of the instances using those properties. Additionally, in the ECS domain we use “ecs:hasName” and “ecs:hasFullName” properties. Then, we rely on the syntactic similarity of (Maedche and Staab, 2002), where the syntactic similarity is inversely proportional to the edit distance:

$$SynSim(L_i, L_j) = \max\left(0, \frac{\min(|L_i|, |L_j|) - ed(L_i, L_j)}{\min(|L_i|, |L_j|)}\right) \in [0,1] \tag{8-2}$$

$SynSim$ returns a degree of similarity between 0 and 1, where two lexical entries L_i and L_j are compared according to edit distance (consider the number of changes needed to change one string into the other) and normalized against the length of shortest string of these two. If two entries match, then the similarity is 1. Assume, we compare two semantic instances, I_i and I_j . I_i has title “Building and managing personalized semantic portals” and is a publication in ECS. I_j has title “Building and Managing Personalized Semantic Portals” and is a publication in the DBLP. As can be seen, both instances refer to same object, but their titles are slightly different because of the use of capital letters. In the example above, we compute $SynSim(I_i, I_j) = 47/51 = 0.92$. In SemWeB, if $SynSim(I_i, I_j) > 0.9$, then we assume two instances are same. Syntactic similarity performs well when same word appears in plural form or written in capital letters or written mistakenly.

8.9.5.2 Ontology-Based Metadata

If the titles of the two compared instances are not syntactically the same, then we compare ontology-based metadata. For this purpose, we weighted the shared edges (i.e. $edge\_weight$ in equation 8-3) between two instances according to the ontological relationships. In our algorithm, different edges are weighted differently as shown below:

$$
\begin{array}{l}
edge\_weight(I_i, p, I_j) = 0 \ \{if\ (p = owl:sameAs)\} \\
edge\_weight(I_j, p, I_i) = 0 \ \{if\ (p = owl:sameAs)\} \\
edge\_weight(I_i, p, I_j) = 2 \ \{if\ (p = rdfs:seeAlso)\} \\
edge\_weight(I_j, p, I_i) = 2 \ \{if\ (p = rdfs:seeAlso)\} \\
edge\_weight(I_i, p, I_j) = 4 \ \{if,(p == ObjectProperty)\} \\
edge\_weight(I_j, p, I_i) = 4 \ \{if,(p == ObjectProperty)\} \\
edge\_weight(I_i, p_1, p_2, I_j) = 4 \ \{if\ \exists o, (I_i, p_1, o) \text{ and } (I_j, p_2, o)\} \\
edge\_weight(I_i, p_1, p_2, I_j) = 4 \ \{if\ \exists o, (o, p_1, I_i) \text{ and } (o, p_2, I_j)\} \\
otherwise, edge\_weight(I_i, p, I_j) = +\infty
\end{array}
\tag{8-3}
$$

where *edge_weight* represents the weight of the edge that exists between the instance I_i and the instance I_j. When there is an owl:sameAs relationship between the two compared semantic instances, then $edge\_weight = 0$ and we assume that those two instances are same. The rdfs:seeAlso relationships weight higher than other ontology-based relationships in our algorithm, where $edge\_weight = 2$. If there is a direct relationship between the two compared instances, then they should be more related than any other two instances that do not have direct relations. Furthermore, we take into account common shared instances between instances. For instance, co-authors of a publication are more likely to be related than two other people who do not have common publication. As a result, direct relations and common instance relations are weighted 4.

While comparing two instances, we look for the shortest path between them according to the weights defined in equation 8-3. Then the relatedness is inversely proportional to the distance:

$$
rel(I_i, I_j) = \frac{1}{shortest(edge\_weight(I_i, I_j)} \tag{8-4}
$$

Figure 8-34 demonstrates an example $edge\_weight$ calculation and syntactic similarity measurement based on the equations 8-2, 8-3 and 8-4.

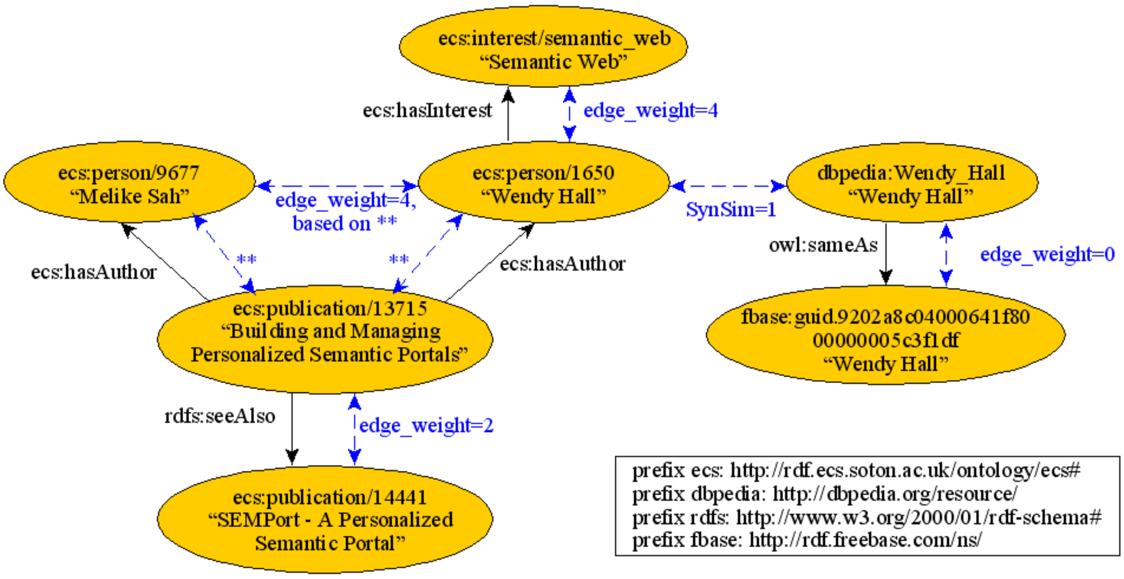


Figure 8-34 Demonstration of $edge\_weight$ calculation

8.9.6 Calculating Semantic Relatedness of a Link to the User Profile

Link annotation is achieved by comparing the semantic relatedness of a link with the user's interests and then by personalizing this measure using the user's interest ranking. In addition, the user's interest rankings may change over time and a time function was used to weight recent interests higher compared to older interests. Thus the resulting similarity measure is user-based. This section summarises the approach we have taken.

The aim is to measure the semantic relatedness of a link to all the interests of the user. For this purpose, a similarity value is calculated between a link and all of the interests of the user as shown in equation 8-5:

$$rld(l,u) = \frac{\sum_{k=1}^{z} rel(l, \text{interest}_k) * rank_k}{\sum_{k=1}^{z} rank_k} \quad (8\text{-}5)$$

where l is a semantic link, u represents the user, interest is the interest of the user to a semantic instance, $rank$ is the interest rating of the interest and z is the no of interests of the user. $rld(l,u) \in [0,1]$ is the relatedness value of a link to the user profile (to all of the interests of the user) and $rel(l, \text{interest}) \in [0,1]$ is the similarity value of the link l to the user's interest interest_k which is calculated by equation 8-4.

Using equation 8-5, the importance of a link to the interests of the user is calculated, where the user's interest rating says how important an interest is to her. However, interests are always changing over time and it is not reasonable to weight all interests the same. For instance, in *"semi-directed"* browsing, current interests of the user are the most significant information for adaptation. In order to weight recent interests higher than previous interests, we used a time function. We have chosen the exponential function for this purpose, since exponential functions do not decay rapidly and over time smoothly decay according to their decay ratio.

$$f(t) = e^{-\lambda t}, \lambda = \frac{1}{\mathrm{T}_0} \tag{8-6}$$

where t represents how many days ago the interest is created , λ indicates decay value and $f(t) \in [0,1]$. The value of $f(t)$ reduces with time. The more recent the interest that is added, the higher is the value of the time function. The exponential function satisfies this condition. Since λ affects decay rate, it is important to select appropriate values for it. Thus, to control the decay rate, T_0 is used. In our adaptation rules, different T_0 values are selected according to different browsing types. Figure 8-35 shows how the time function is changed based on diverse T_0 value. When T_0 has low values, the decay is rapid. Therefore, the more recently added interests of the user are weighted higher. When T_0 has higher values, the decay is slow and all interests contribute with different degrees. When the browsing type is "semi-directed", our aim is to guide users to related links according to their most recent interests and we set $\mathrm{T}_0 = 20$, a small value. When browsing type is "undirected", then we set $\mathrm{T}_0 = 200$, a higher value.

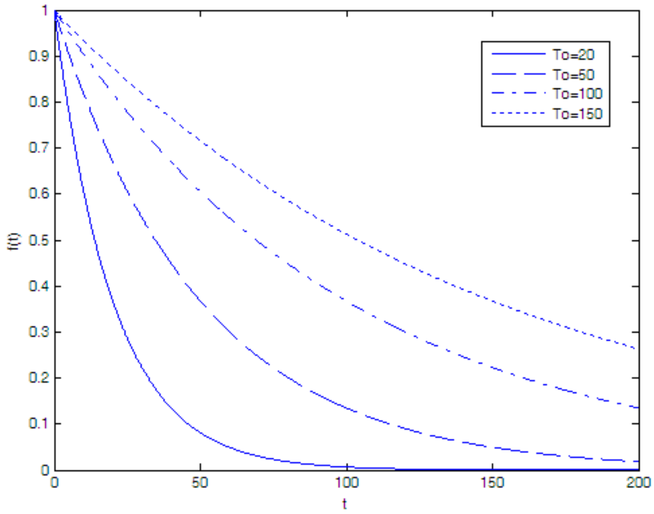


Figure 8-35 Experimenting with different T_0 values

The final user-based semantic relatedness measure with the added time function is shown in equation 8-7. Our semantic relatedness measure is not ontology dependent, therefore can be applied to different ontologies that do not represent taxonomies.

$$rld(l,u) = \frac{\sum_{k=1}^{z} rel(l, \mathrm{interest}_k) * rank_k * f(t_k)}{\sum_{k=1}^{z} rank_k} \tag{8-7}$$

8.9.7 Link Annotation with Visual Cues during Browsing

According to (Brusilovsky, 2004), annotating links (augmenting with extra information) can reduce the number of visited nodes in a hypertext system and hence reduces the learning time. In SemWeB, the interests of the user are used as a basis for the annotation of links with visual cues. As discussed in the previous section, the relatedness of a link to the user profiles (i.e. $rld(l,u)$ in equation 8-7) is calculated using ontological metadata. This measure is also personalized based on the user's ranking (preference). For each created semantic link created by the SLS, l, a semantic relatedness value is calculated using $rld(l,u)$ as described in equation 8-7. Then each relatedness value is normalized by the link that has the maximum relatedness value:

$$\sum_{i}^{m} rld_{nor}(l_i,u) = \frac{rld(l_i,u)}{\max(rld(l,u))} \tag{8-8}$$

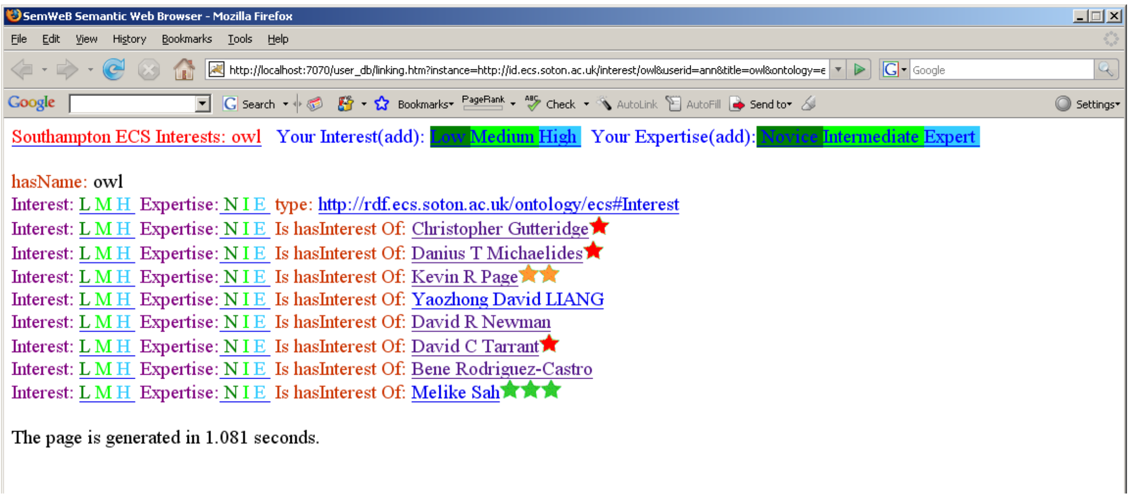
where l_i is a semantic link generated by the SLS, m represents the number of semantic links created by the SLS, u is the user and $\max(rld(l,u))$ represents the semantic relatedness value of a link l that has the maximum value among all the SLS created links. The final normalized similarity value is thresholded and different visual cues are added to the links during the presentation to the user using equation 8-9. In the tests it was observed that these thresholding values worked better compared to other thresholds.

$$\begin{array}{l} \text{if } rld(l_i,u) >= 0.75, \text{annotate link with three green stars} \\ \text{if } rld(l_i,u) >= 0.5, \text{annotate link with two orange stars} \\ \text{if } rld(l_i,u) >= 0.25, \text{annotate link with a red star} \\ \text{otherwise, no link annotation} \end{array} \tag{8-9}$$

A demonstration of link annotation: Assume Ann added a few interests to her profile as shown in Figure 8-36 (a) such that she has medium interest in “Melike Sah”, and high interest in “Semantic Web”, “Ontologies” and “Semantic Annotation”. Then, during browsing, she requested the “owl” instance from SemWeB. As shown in Figure 8-36 (b), the presented hyperlinks are annotated with different visual cues depending on her interests. It is also noted that the same Web page was generated within 0.14 secs without adaptation and with adaptation Web page generation time was increased to 1.081 secs, where we use a Windows XP operating PC with 2.87 GB RAM and 3.20 GHz CPU.

Attribute	Value
Color Scheme Preference	http://localhost:7070/user_db/user_schema.owl#bluescheme Change
Email	ann@gmail.com Change
Name	Ann Change
Has Interest:	http://id.ecs.soton.ac.uk/person/9677 Value: Medium
Has Interest:	http://id.ecs.soton.ac.uk/interest/semantic_web Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/semantic_annotation Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/ontologies Value: High

(a)



(b)

Figure 8-36 (a) Ann's browsing interests. (b) Recommended semantic hyperlinks using link annotation according to Ann's profile

Here, we would like to demonstrate how the created links are annotated using our proposed semantic relatedness measure. In our algorithm the similarities of the semantic links created by the SLS are compared with all of the interests of the user. According to the user's interests, the shortest semantic distance between a user interest and the created semantic hyperlink is found using the edge weights defined in equation 8-3. These distances are shown in Figure 8-37 according to the example. As can be seen, the "Melike Sah" instance has the most shared edges with Ann's interests compared to other instances. Then, according to the interest values (i.e. medium), we calculated the following semantic relatedness measures, $rld(l,u)$, to each semantic hyperlink using equation 8-7: ((2*1+3*0.25+3*0.25+3*0.25)/11=0.38) for Melike Sah, ((3*0.25+3*0.25+3*0.25)/11 =0.20) for Kevin R Page, ((3*0.25+3*0.25)/11=0.13) for Danius Michaelides and ((3*0.25+3*0.25)/11=0.13) for Christopher Gutteridge. Because all of the interests are created within the browsing session (i.e. t=0), the time function did not have any affect on the final similarity values in this case.

The similarity values are added to the response XML using the "similarity" tag defined by the presentation vocabulary. In addition, the highest similarity value is added to the response using the "MaxSimilarity" tag. At the browser, similarity values are normalized by the highest similarity value, which is 0.38 in this case and annotated with different visual cues based on equation 8-9 and presented as shown in Figure 8-36 (b).

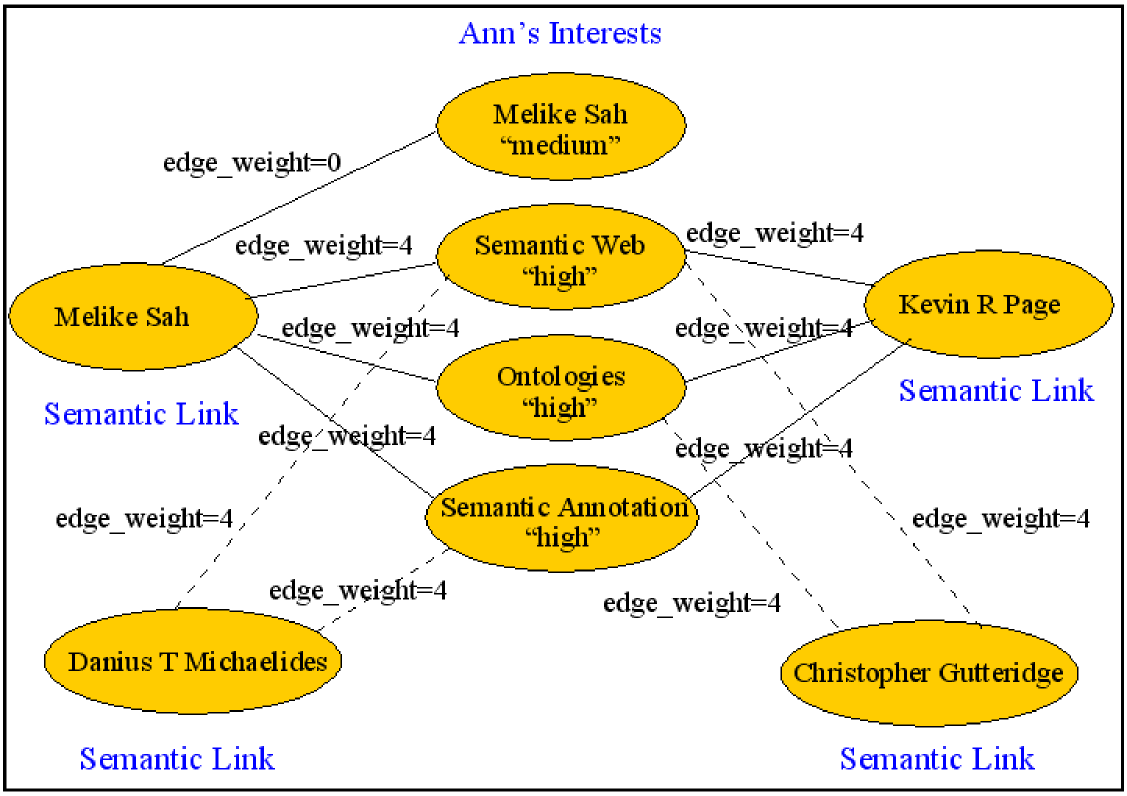


Figure 8-37 Demonstration of semantic distances between Ann's interests and semantic hyperlinks according to the equation 8-3

8.10 Adaptive Text Presentation and Link Creation Based on the User's Expertise

In SemWeB, adaptive text and links are generated according to the expertise of the user. In the study of Carmel et al. 1992, they found that users with different expertise should be treated differently. For instance, novice users in an area need explanations of the concept and referential links. On the other hand, expert users require detailed links to other Web resources. To provide expertise-based adaptation to different users, in the proposed user model we have introduced the expertise concept, where users can indicate their experience in different semantic instances on the Web (i.e. novice, intermediate and expert).

Depending on the expertise of the user on a concept, different adaptation is provided. For instance, if the user has entered an expertise value of *"novice" or "intermediate"* for a semantic resource, then while presenting this resource to the user, we automatically generate adaptive text to the Wikipedia definition of the resource using the DBD goal service and more links to the related Web resources within Wikipedia are presented using the FMLDB goal service. If the user is expert, then we show more links to the related Wikipedia pages using the FMLDB goal service. In this way, the user is guided to related Web resources automatically within browsing session. The algorithm in Figure 8-38 is used to adapt text and links based on different expertise values.

```
@prefix user_schema:http://localhost:7070/user_db/user_schema.owl#
When(?user user_schema:HasExpertise ?x), (?x user_schema:About ?uri),
((?x, user_schema:HasExpertiseValue user_schema:novice) ||
(?x, user_schema:HasExpertiseValue user_schema:intermediate)) -> invoke DBD and FMLDB

When(?user user_schema:HasExpertise ?x), (?x user_schema:About ?uri),
(?x, user_schema:HasExpertiseValue user_schema:expert) -> invoke FMLDB
```

Figure 8-38 Algorithm to generate adaptive text and links according to different expertise values

8.11 Personalized Homepages

Users surf around on many Web pages on the Web. A personalized homepage may help them to locate their favourite Web pages, such as myYahoo and iGoogle provide to their users. In SemWeB, we also support a personalized homepage, where we provide links to the added browsing interests of the users and more links to related resources. The links are ordered according to the interest creation time and related links are annotated with visual cues according to the proposed semantic relatedness value that we discussed in section 8.9. Thus, the homepage also supports AH and provides access to part of the information on the Web according to the user's choices. A screen shot of the personalized homepage is shown in Figure 8-39. The personalized homepage can be accessed from the personalization tab of the SemWeB sidebar.

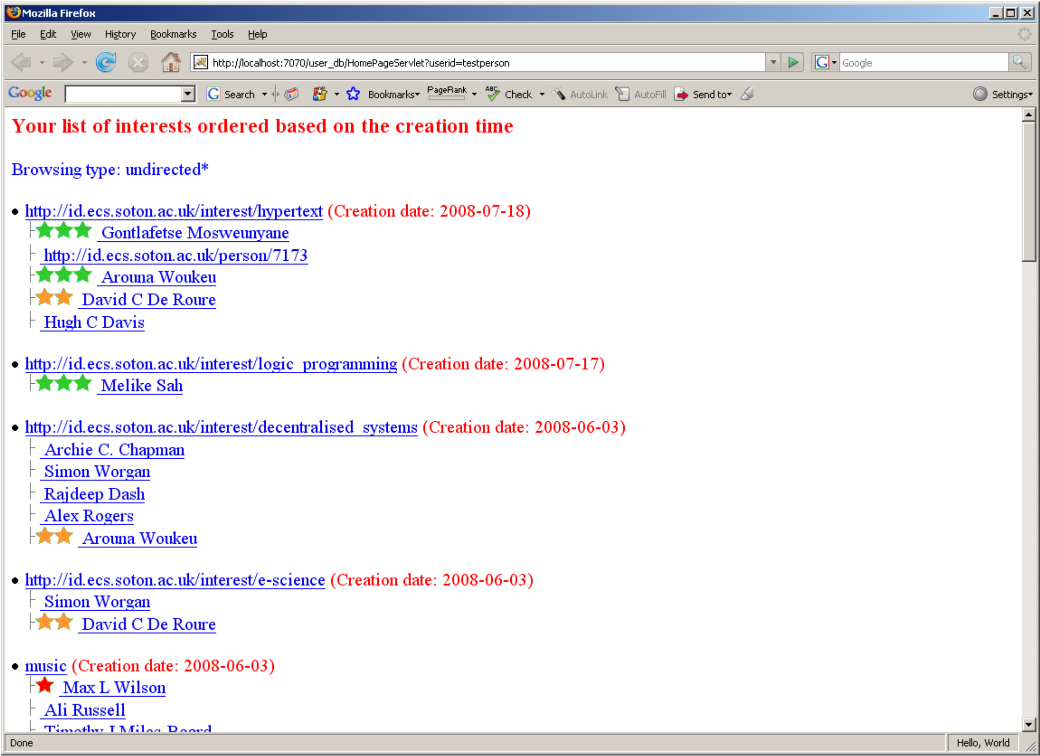


Figure 8-39 Personalized homepage of the user

8.12 Chapter Summary

This chapter discussed the system design, our semantic annotation approach, the proposed user model, the user modelling, the context-based link creation, the goal services and the personalization features of the proposed Semantic Web browser, SemWeB in detail. In our approach, we supported open-corpus semantic linking and AH on different Web domains. For enabling context-based hyperlink creation on different Web domains, we perform semantic annotation using a linked data domain and in this chapter, we tested SemWeB on the ECS domain. In addition, we illustrated ontology-based link creation using dereferenceable URIs and purposefully developed goal services. To achieve open-corpus AH on different domains, we proposed a behaviour-based and ontology-driven user model, and we demonstrated how AH can be supported using our proposed user model in the ECS domain. In addition, we developed a new user-based semantic relatedness measure for link annotation with visual cues and explained it in detail. In the next chapter, we discuss system-based and scenario-based evaluations of SemWeB.

9 Evaluation of SemWeB

We carried out a system-based evaluation and a scenario-based evaluation of the proposed Semantic Web browser, SemWeB. This chapter explains the evaluations that were undertaken.

9.1 System-Based Evaluation

A number of evaluations were performed to test the adaptability and genericity of SemWeB to different ontologies and URIs. The scalability of the SemWeB system architecture is also discussed. The rest of this section summarises the research undertaken.

9.1.1 Genericity

Genericity is an important measure, which tests if a system can work on different domains. We tested SemWeB on different ontologies and URIs to show that it is not an application-specific software. In addition to the ECS linked data domain, we have tested SemWeB on all of the instances of the DBpedia (DBpedia, 2008) and on a small set of URIs from the DBLP (DBLP, 2008).

Genericity of Semantic Annotation: In order to test different linked data domains on SemWeB, we need to make some changes to the IE and semantic annotation module. For instance, we need to make changes to the modified GATE framework, such as new gazetteers are needed to be created depending on the linked data domain, new dereferenceable URIs are required to be added to the mapping database, new JAPEC rules are required or existing rules need to be modified and the SemWeB sidebar needs to be extended with the new ontological class names according to the ontology.

Genericity of Semantic Link Generation: The SLS works without any updates since it is application-independent and can make use of any valid dereferenceable URI for creating semantic links. In the same way, the SLS-based generic goal services can be re-used by other domains, such as the "Finding DBpedia Definition" goal service.

Genericity of User Modelling: The proposed user model ontology is independent of ontologies and it is designed to work with diverse domains. For instance, users can add interests and expertise to any dereferenceable URI independent of ontologies.

Genericity of Adaptation: Our proposed adaptation approach can work in different domains without modification. For instance, the proposed semantic relatedness measure is ontology-independent, generic goal services can be tested on different domains and expertise-based adaptation again does not require any updates.

9.1.1.1 Experimentation of SemWeB using DBpedia

DBpedia extracts structured information from Wikipedia and makes it available in the form of RDF on the Web (DBpedia, 2008). As of November 2008 (DBpedia, 2008), DBpedia dataset version 3.2 describes about "2,600,000" things, including information about many world objects, such as people, places, films, etc. In addition, it contains 609,000 links to images, 3,150,000 links to external Web pages and 4,878,100 external links into other RDF datasets; such as Geonames, MusicBrianz, DBLP. Also, DBpedia contains disambiguation links, which are very useful, if an instance has more than one meaning, users can be guided to the right resources using DBpedia RDF links. As a conclusion, the DBpedia dataset provides very useful metadata and it can be used to create ontology-based hyperlinks on Web pages to guide users to relevant information and links.

The Procedure Used: For experimenting with DBpedia URIs on the SemWeB, we have used all DBpedia instances from DBpedia version 3.2. To generate gazetteers for IE, we only need instance names (lexicons). For this purpose, we have used instance titles in English, which is available to download from the DBpedia downloads Website in N-triple format[60]. Then, we parsed this file with Jena and extracted instance URIs and instance lexicons (rdfs:label annotation property is used) using SPARQL queries. For DBpedia mappings, we have created a new mapping database, called *DBpedia mappings*. Each extracted instance URI and lexicon was then added to the DBpedia mappings database. We also created gazetteers from the extracted lexicons. It took approximately 9 hours to extract the URIs and lexicons, in the end 2,721,702 URI-lexicon mappings were stored to the DBpedia mappings database. Since the mapping database is very big, query times were very slow; on average it took 25-30 seconds to answer each query. To improve query times, we indexed the DBpedia mapping database and currently on average it takes 0.02-0.03 seconds to query this database. Furthermore, we used the extracted lexicons to create gazetteers for IE within GATE. We divided the lexicons to 10 gazetteer files, since it was taking approximately 1 minute to load one big gazetteer containing all DBpedia instances. We also increased the Web server (i.e. Tomcat) cache memory to be able to perform annotations faster. The reader is referred to sections 8.4.1 and 9.1.3.2 for more scalability discussion of semantic annotation.

Based on the gazetteers that we created for DBpedia domain, new JAPEC rules were generated (created JAPEC rules are shown in Figure A-14 in Appendix A). After creating the necessary components for IE and semantic annotation, the last step was to add the corresponding DBpedia ontological classes to the SemWeB sidebar code. Instead of showing the whole DBpedia class hierarchy in the SemWeB sidebar, we adopted a different presentation approach. The DBpedia class hierarchy is too big to show in the sidebar. Instead, we only show one link in the sidebar, which we called *"DBpedia Links"*. When the user selects this check box, ontology-based links to the DBpedia instances are added to the Web page. However, DBpedia has many instances, including stop words, such as "homepage", "view", "business", "contact", etc. and as a result, too many links were embedded when this link was selected. To solve this problem, we filtered DBpedia instances that are important within the English Wikipedia. This is achieved by checking idf (inverse document frequency) of each semantic instance recognized during the semantic annotation process. idf measures the importance of a word in the whole corpus of documents as shown in equation 10.1

[60] http://wiki.dbpedia.org/Downloads

and it acts as a link filter during the semantic annotation. In addition to idf, we have experimented with filtering DBpedia instances using $tf \times idf$ (term frequency, inverse document frequency), to stress DBpedia instances particularly important to the document. We have observed that if a DBpedia instance has a very low idf value, such as "–infinite", then the tf value does not make a difference on the final filtering. Since calculating tf adds some delay to the semantic annotation and it does not make much difference, we removed tf from the filter.

$$idf_i = \log \frac{|D|}{\left|\{d_j : t_j \in d_j\}\right|} \tag{10.1}$$

where, $|D|$ is the total number of documents in the corpus. In this case, it is the total number of documents within the English Wikipedia. To estimate the number of documents in the English Wikipedia, we have queried the English Wikipedia using the Yahoo Search API[61] and searched for "Wikipedia" term in the "en.wikipedia.org" domain (i.e. every Wikipedia page has this keyword in the document). The search query returned 17,700,000 results by the date 29 May 2009, where this is the number of pages Yahoo indexes in the English Wikipedia and we have used this number as the value of $|D|$. $\{d_j : t_j \in d_j\}$ is the total number of documents where term t_j appears. In this case, t_j is the lexicon of a DBpedia instance that is recognized from the Web page.

To calculate the inverse document frequency, we used the Yahoo Search API. This allows a term to be searched for inside a Web site domain. During the semantic annotation, the lexicons of the recognized DBpedia instances are searched within the Wikipedia English Website (entire corpus for DBpedia) and the number of occurrences within the entire corpus is found. For instance, the following HTTP request is made to the Yahoo Search API with the lexicon and the site restriction parameters.

```
http://search.yahooapis.com/WebSearchService/V1/webSearch?appid=YahooDemo&query=%22"+
lexicon+"%22&site=en.wikipedia.org
```

Figure 9-1 HTTP Yahoo search request using Yahoo search API

[61] http://search.yahooapis.com/WebSearchService/V1/webSearch [last accessed, 5/11/2008]

The search results are sent back in XML by the Yahoo server. We parse this XML response and obtain the total matches of the searched lexicon within the Wikipedia Website. This value is utilized in the calculation of idf. Then, we filter DBpedia instances that have $idf > 5.0$. We have experimented with different threshold values for idf. For common words, such as homepage, please, etc. usually $idf = -\text{infinite}$. When $idf > 3.0$, we observed that most of the stop words also appeared in the final semantic annotation. After experimenting, we selected $idf > 5.0$. However, the decision of selecting the granularity of links can be given to the user from the SemWeB sidebar.

We have implemented idf within IES. When the selected ontology is DBpedia, then while creating semantic annotations in the IES, we also find the idf measure by invoking the Yahoo Search API using HTTP requests. Then, if $idf > 5.0$, the DBpedia instance is added to the semantic annotation, otherwise it is not included in the annotation. An example annotation using DBpedia is shown in Figure 9-2. In addition, in the Figures 9-3, 9-4 and 9-5, we demonstrate semantic annotation with different idf values. As the threshold increases, the selectivity also increases.

```
<?xml version="1.0"?> <message>
 <DBpediaClass>
   <value>Milky Way</value>
   <mapping> http://dbpedia.org/resource/Milky_Way</mapping>
 </DBpediaClass>
</message>
```

Figure 9-2 An example SemWeB annotation using the DBpedia

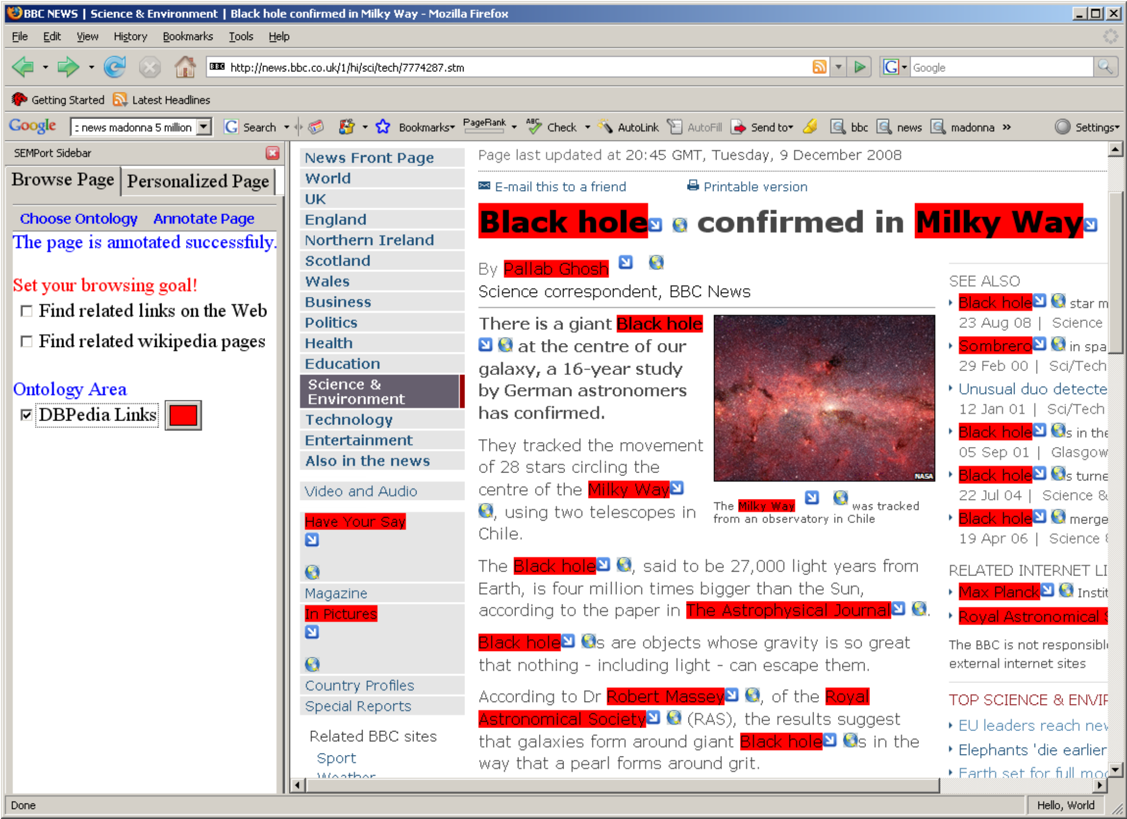


Figure 9-3 An example semantic annotation using DBpedia, where *idf* >= 7.0

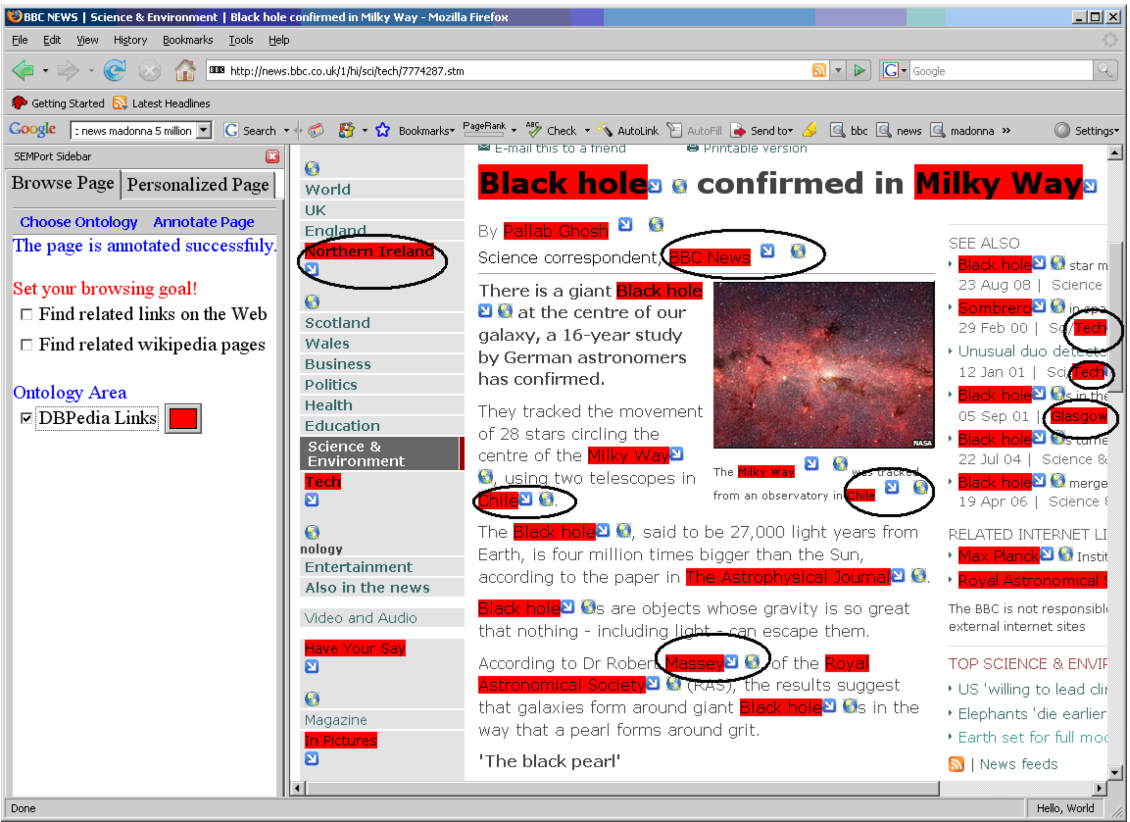


Figure 9-4 An example semantic annotation using DBpedia, where *idf* >= 5.0[62]

---

[62] The circled DBpedia instances are the newly added instances comparing to *idf* >= 7.0

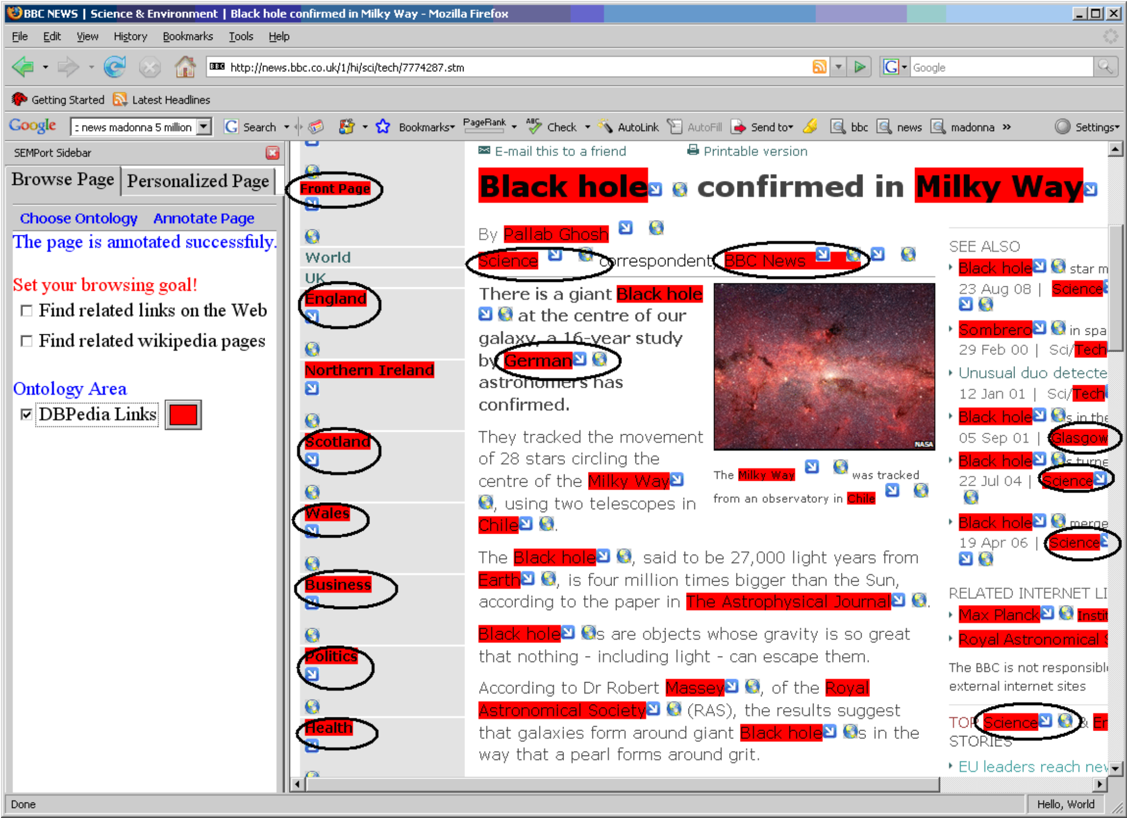


Figure 9-5 An example semantic annotation using DBpedia, where *idf* >= 3.0[63]

Using DBpedia in the SemWeB Sidebar: In order for users to use DBpedia for semantic link generation, they need to choose the DBPedia ontology from the navigation tab of SemWeB sidebar extension. Then, they can annotate the Web page using instances from this domain. IES extracts semantic instances based on the DBpedia gazetteers and annotates the Web page according to JAPEC rules provided in GATE. Using the annotated Web page, users can highlight the relevant recognized DBpedia instances on the Web page as shown in Figures 9-3, 9-4 and 9-5. When the user clicks one of the embedded links, more information and semantic links are generated by the SLS as shown in Figure 9-6. Since SLS is ontology and URI independent, any DBpedia URI can be used for creating semantic information and links. In the same way as for the ECS domain, we used lexicons of the instances as anchor names while presenting links.

[63] The circled DBpedia instances are the newly added instances comparing to *idf* >= 5.0

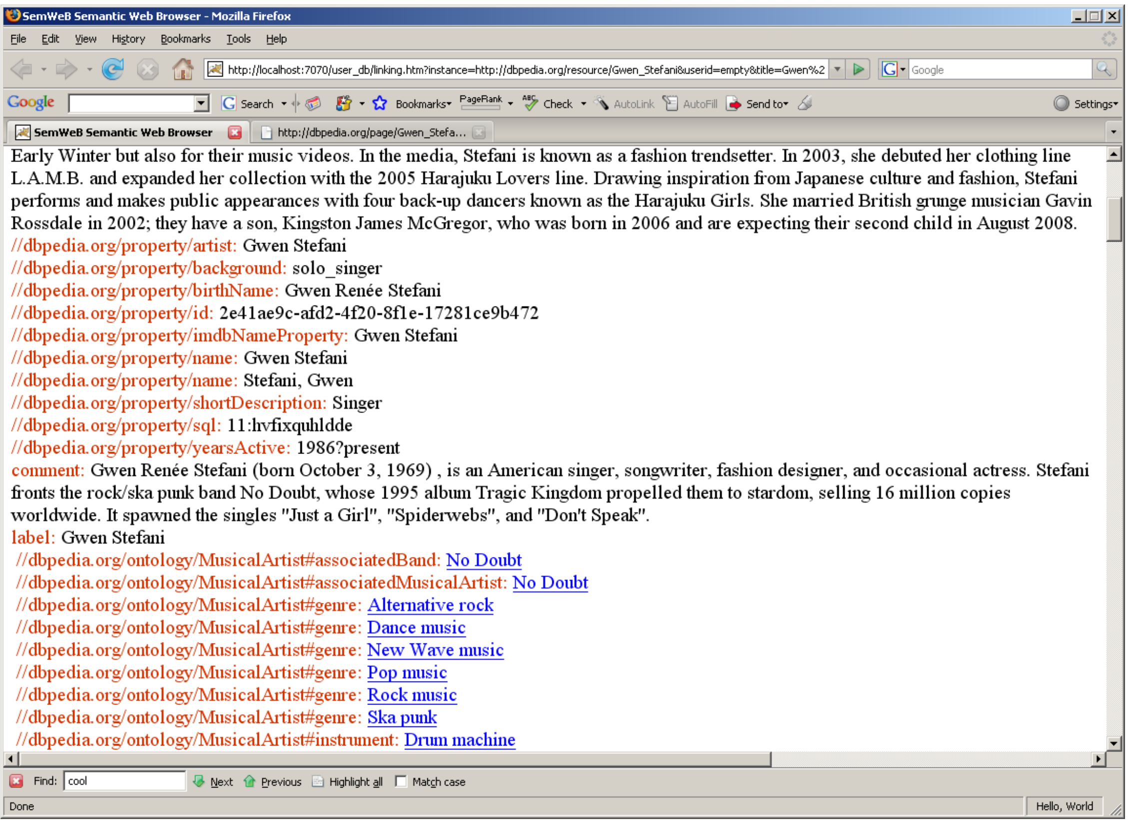


Figure 9-6 Semantic links created by SLS using a DBpedia URI

Testing Genericity of Personalization on DBpedia: To provide open-corpus AH on different domains, we have used ontologies. To test the genericity of the personalization, we tested our user model ontology, goal services and proposed semantic relatedness measure on the DBpedia domain and this section illustrates the experiments that were carried out.

Testing the User Model Ontology on DBpedia: In the proposed user model ontology, user profiles can point to any dereferenceable URI from diverse datasets. In Figure 9-7, we showed that user profiles can be extended with different URIs. An example user profile which contains URIs from different datasets is also shown in Figure 8-11.

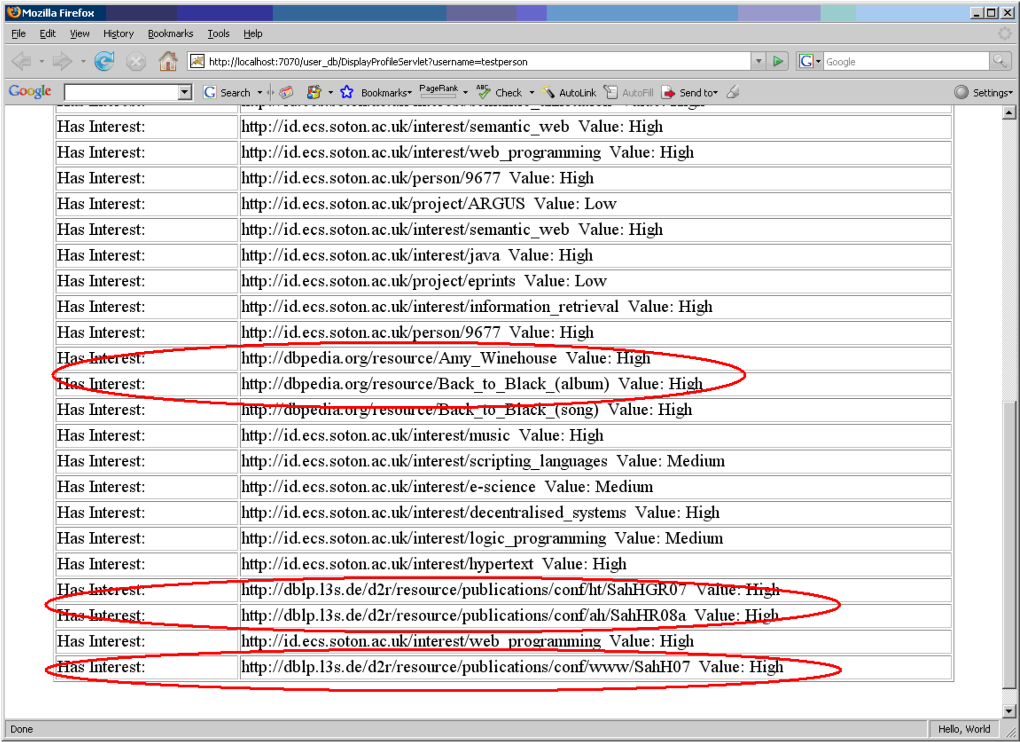


Figure 9-7 An example user profile that points to diverse linked data URIs on the Web

Testing Goal Services on DBpedia: SemWeB generic goal services can be applied to different ontologies. In DBpedia, we have tested the "Searching for Related Semantic Information on the Web" (SRSIW) and "Find More Links within DBpedia" (FMLDB) goal services. FMLDB provides links to "skos:subject" and "skos:broader" resources and in the RDF description of a DBpedia instance, these links are already included. Therefore, we modified FMLDB for the DBpedia domain as shown in Figure 9-8 such that we find "skos:broader" topics of the "skos:subject" and "skos:broader" topics of a DBpedia instance. For instance in Figure 9-9, the links created by SRSIW and the modified FMLDB are shown (i.e. the user requested "http://dbpedia.org/resource/World_Wide_Web" DBpedia instance). In addition, more specific goal services can be designed especially for DBpedia in future work.

```
CONSTRUCT
{<"+DBpedia_URI+"> <http://www.w3.org/2004/02/skos/core#related> ?subject.}
WHERE
{<"+DBpedia_URI+"> <http://www.w3.org/2004/02/skos/core#subject> ?z.
?subject <http://www.w3.org/2004/02/skos/core#broader> ?z.}
UNION
{<"+DBpedia_URI+"> <http://www.w3.org/2004/02/skos/core#broader> ?z.
?subject <http://www.w3.org/2004/02/skos/core#broader> ?z.}}
```

Figure 9-8 Modified SPARQL query to find more related links within DBpedia

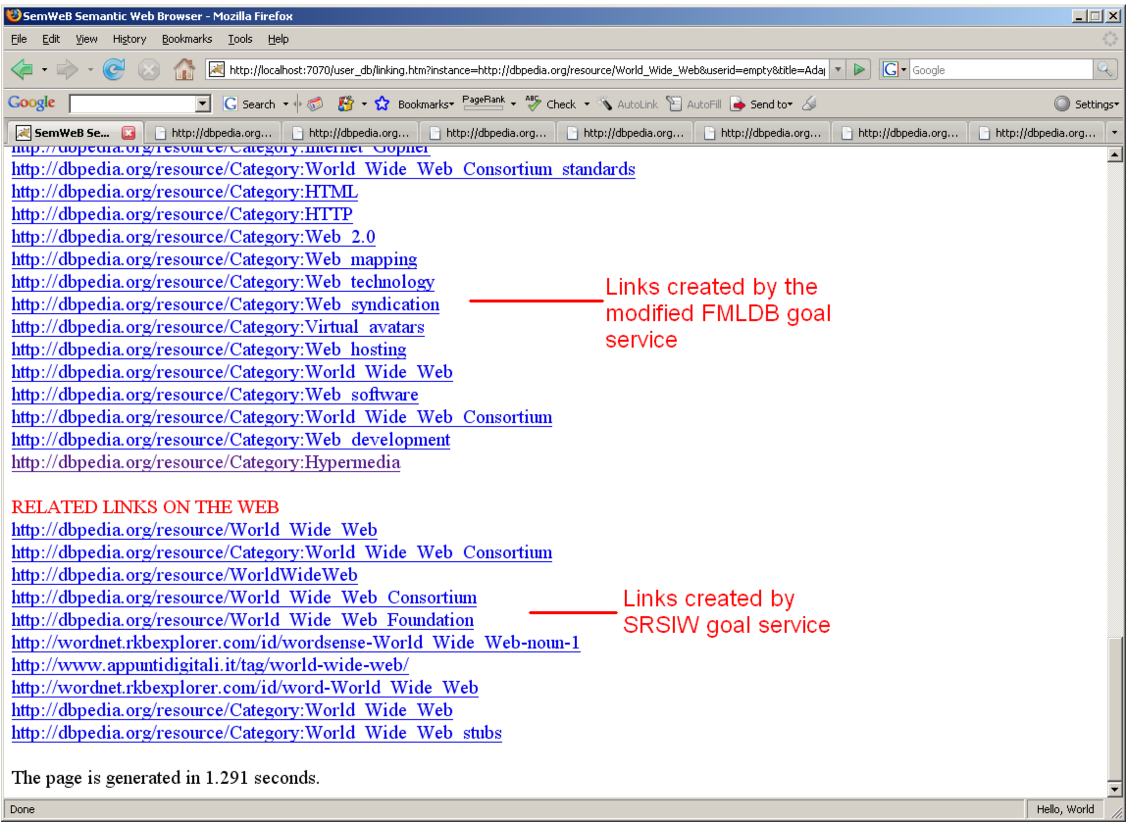
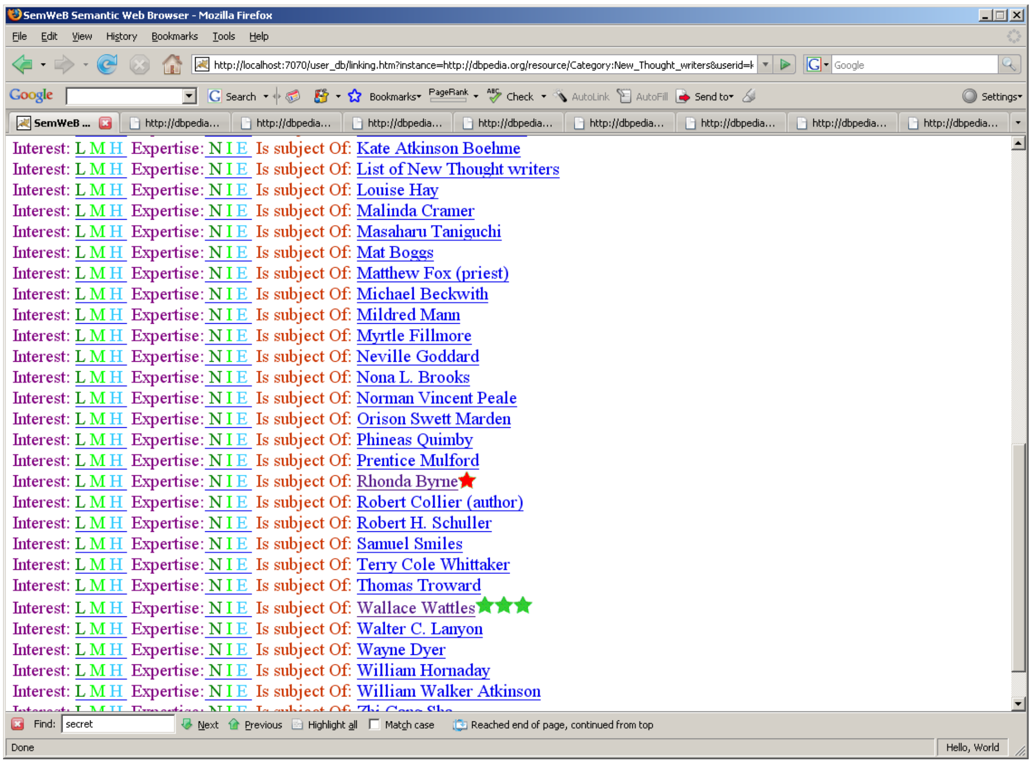


Figure 9-9 Demonstrations of SRSIW and FMLDB goal services in DBpedia

Testing the Semantic Relatedness Measure on DBpedia: The proposed semantic relatedness measure can be used on different ontologies. After adding some DBpedia interests to the user profile as shown in Figure 9-10(a), some of the DBpedia links are annotated with visual cues by the adaptation module as illustrated in Figure 9-10 (b). For instance, "Rhonda Byrne" is recommended with one red star during browsing using the proposed semantic relatedness measure, since she is the producer of the one of the user's interests (i.e. The Secret movie). Author "Wallace Wattles" is also recommended since the user has added this resource to the profile.

Attribute	Value
Color Scheme Preference	http://localhost:7070/user_db/user_schema.owl#greensheme Change
Name	katie Change
Has Interest:	http://dbpedia.org/resource/Wallace_Wattles Value: High
Has Interest:	http://dbpedia.org/resource/Category:New_Thought_writers Value: High
Has Interest:	http://dbpedia.org/resource/Category:Self-help_books Value: High
Has Interest:	http://dbpedia.org/resource/The_Science_of_Getting_Rich Value: High
Has Interest:	http://dbpedia.org/resource/The_Secret_(2006_film) Value: High

(a) An example user profile



(b) Annotated Web page with different visual cues

Figure 9-10 Demonstration of link annotation with visual cues in the DBpedia domain

9.1.1.2 Experimentation of SemWeB using DBLP

The DBLP database provides bibliographic information on the main computer science journals and conferences in the form of RDF on the Web (DBLP, 2008). It contains more than 950,000 articles and 570,000 authors. In research or education-related Web pages, DBLP can be used to provide ontology-based links to relevant resources.

The Procedure Used: We tested a small set of URIs from the DBLP database. We manually stored a set of DBLP URI-lexicon mappings to a new database named *DBLP mapping*. In addition, new gazetteers and JAPEC rules were created. Finally, we extended the SemWeB sidebar with the ontological classes from the DBLP ontology, such as Author, Journal, Publication, Conference and Collection. The SLS and adaptation module work without modifications on the DBLP domain.

Using the DBLP in the SemWeB Sidebar: Users can use the DBLP ontology by selecting from the navigation tab of the SemWeB sidebar extension. When the user choses to annotate the Web page, the IES uses DBLP gazetteers, JAPEC rules and DBLP mappings database to extract and annotate DBLP instances from the Web page.

In Figure 9-11, an example SemWeB annotation using the DBLP domain is shown. Using the annotated Web page, users can highlight and embed links to the recognized DBLP instances on the Web page as presented in Figure 9-12. When the user clicks on the embedded link, the SLS creates ontology-based information and links, and presents this information in a new Web page as shown in Figure 9-13. The mapping database is used to find lexicons of the URIs and these lexicons are utilized to present links with user-friendly anchors.


```
<?xml version="1.0"?> <message>
<Author>
  <value>Melike Sah</value>
  <mapping>http://dblp.L3S.de/d2r/resource/authors/Melike_Sah</mapping>
</Author>
<Publication>
  <value>Designing a Personalized Semantic Web Browser.</value>
  <mapping>http://dblp.l3s.de/d2r/resource/publications/conf/ah/SahHR08</mapping>
</Publication>
</message>
```


Figure 9-11 An example SemWeB annotation using the DBLP

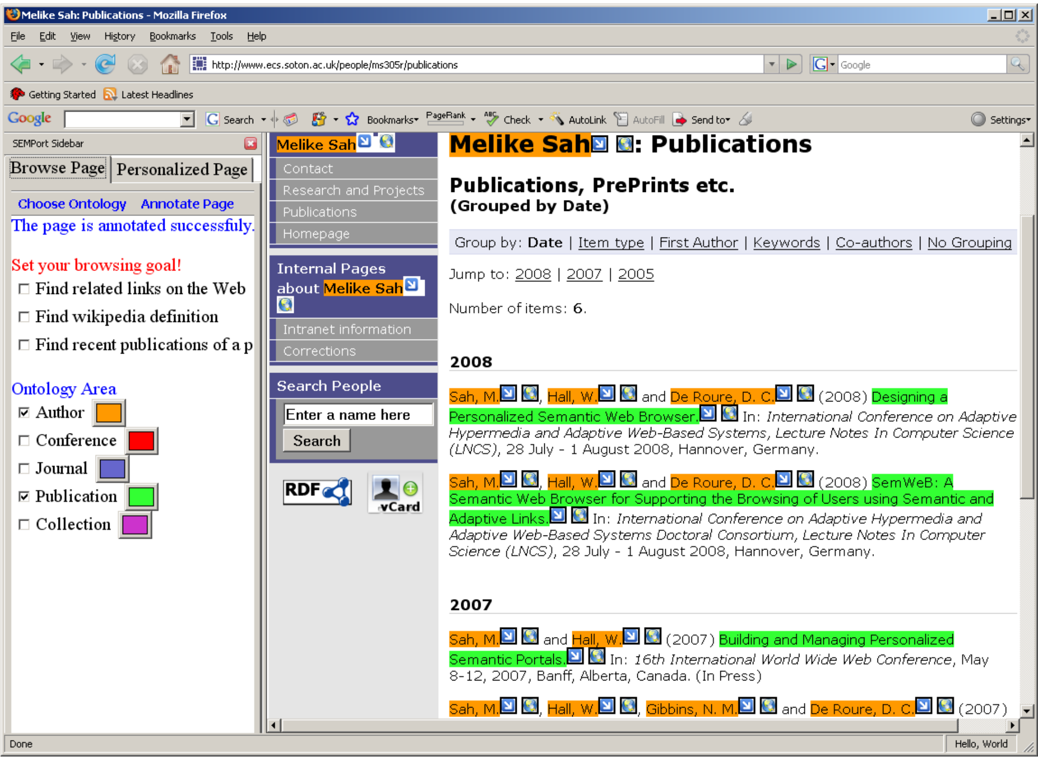


Figure 9-12 SemWeB browser with the embedded DBLP hyperlinks

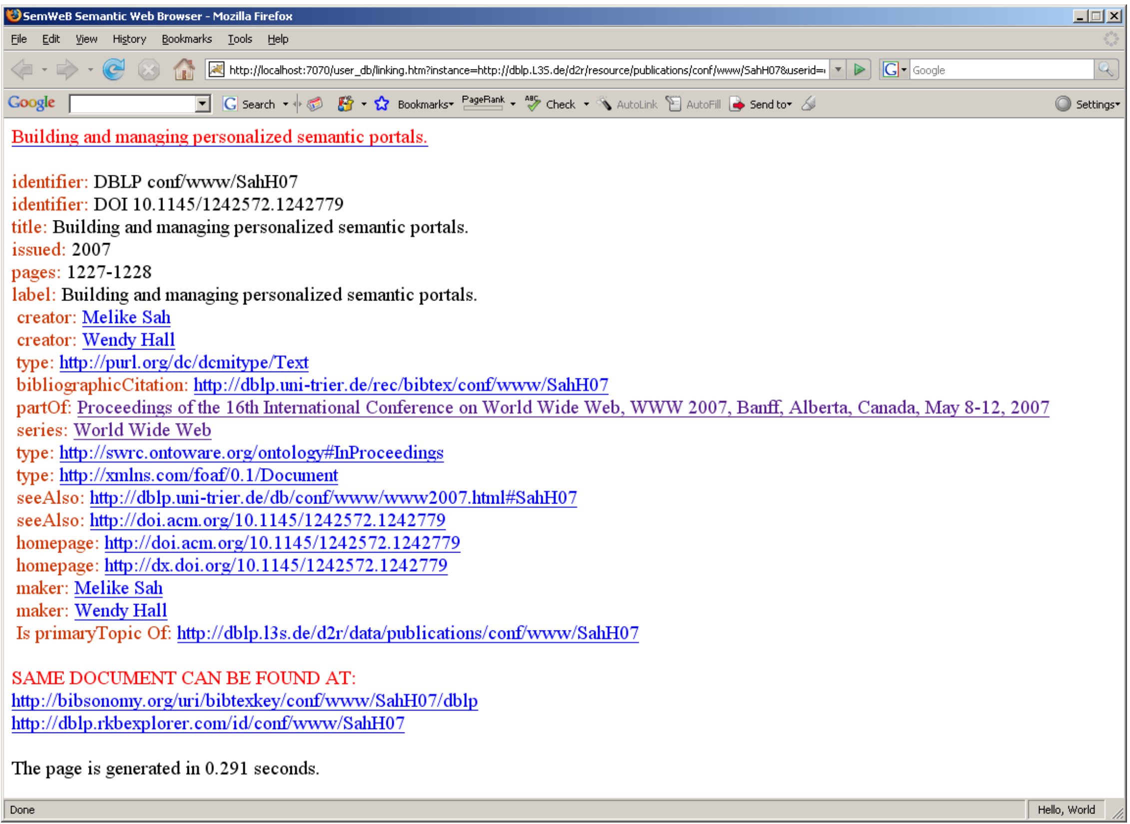
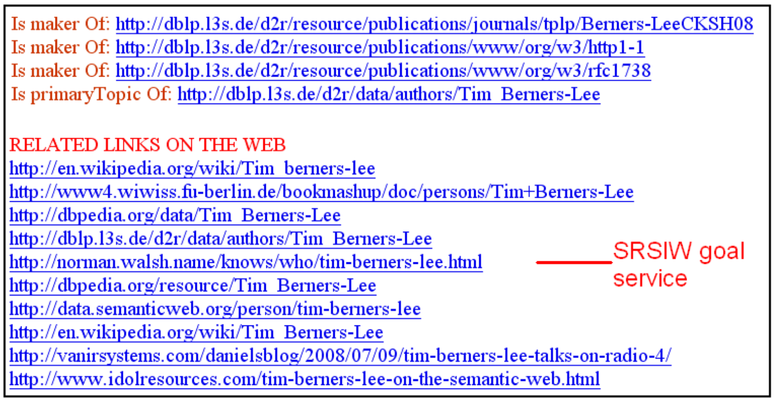


Figure 9-13 Created semantic links and information by SLS using a DBLP URI

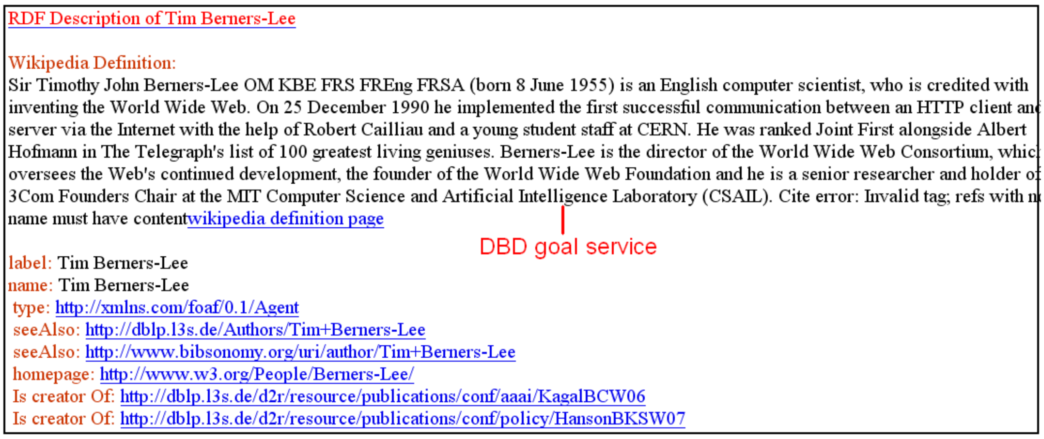
Testing Genericity of Personalization on the DBLP: We have experimented with the genericity of our user model ontology, goal services and the proposed semantic relatedness measure on the DBLP domain and this section presents a number of tests that were undertaken.

Testing the User Model Ontology on DBLP: Our ontology can point to any dereferenceable URI as shown in Figure 9-7.

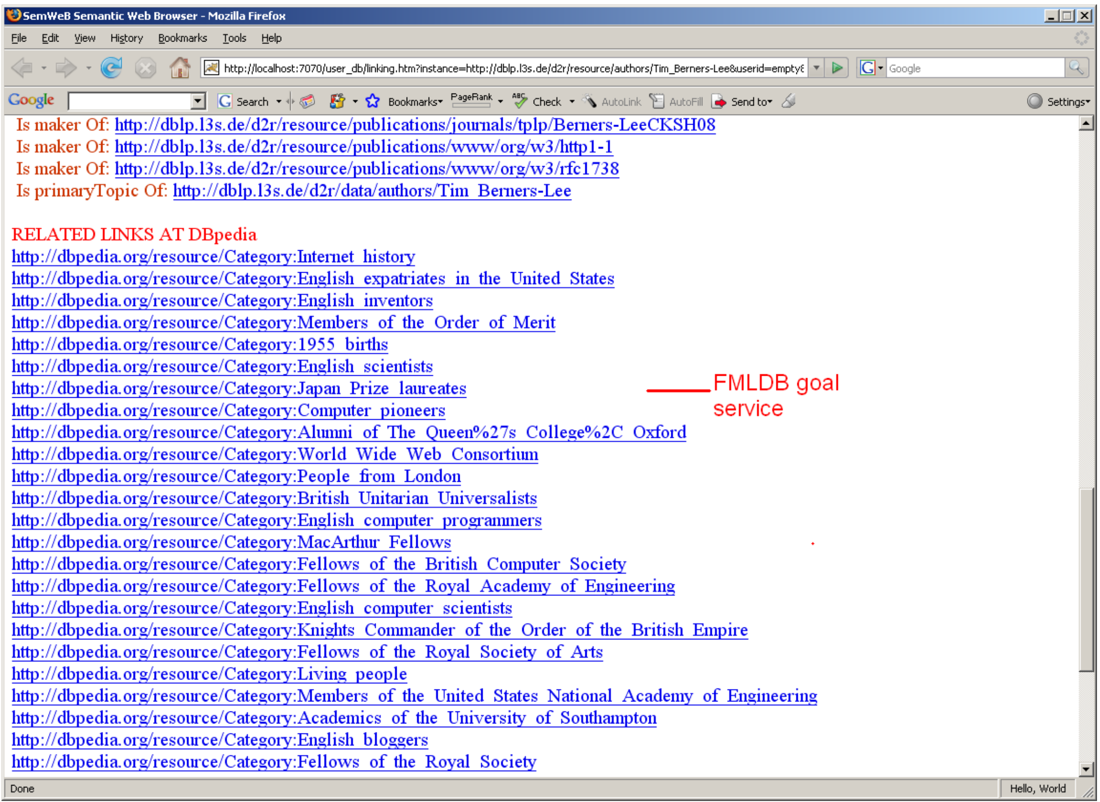
Testing SemWeB Goal Services on DBLP: We tested the genericity of generic goal services. "DBpedia Definition" (DBD), "Search for Semantic Related Information on the Web" (SRSIW) and "Find More Links within DBpedia" (FMLDB) were tested on the DBLP domain as shown in Figure 9-14. The experiments showed that different services can be used on the DBLP. In addition, in future work specific goal services to the DBLP domain could be generated to guide users to related contents, such as related conferences or related publications.



(a) SRSIW goal service in the DBLP



(b) DBD goal service in the DBLP



(c) FMLDB goal service in the DBLP

Figure 9-14 Tested goal services in the DBLP domain

Testing Semantic Relatedness Measure on DBLP: Some tests were done to show ontology-based personalized links in the DBLP domain as shown in Figure 9-15. For instance, we added the publication "SEMPort personalized semantic portal", "Building and Managing Personalized Semantic Portals" and "SemWeB: A Semantic Web Browser for Supporting Browsing of Users using Semantic and Adaptive Links" to the interests of the user as shown in Figure 9-7. Note that, "Melike Şah" and "Wendy Hall" appeared as authors of all three interests, and "David De Roure" appeared as author of two of the interests. As demonstrated in Figure 9-15, "Melike Şah" and "Wendy Hall" are recommended to the user with three stars. "David De Roure" is recommended with two stars and also since "Adaptive Hypermedia conference" is related to the one of the interests, it also recommended with one star by using the proposed semantic relatedness measure.

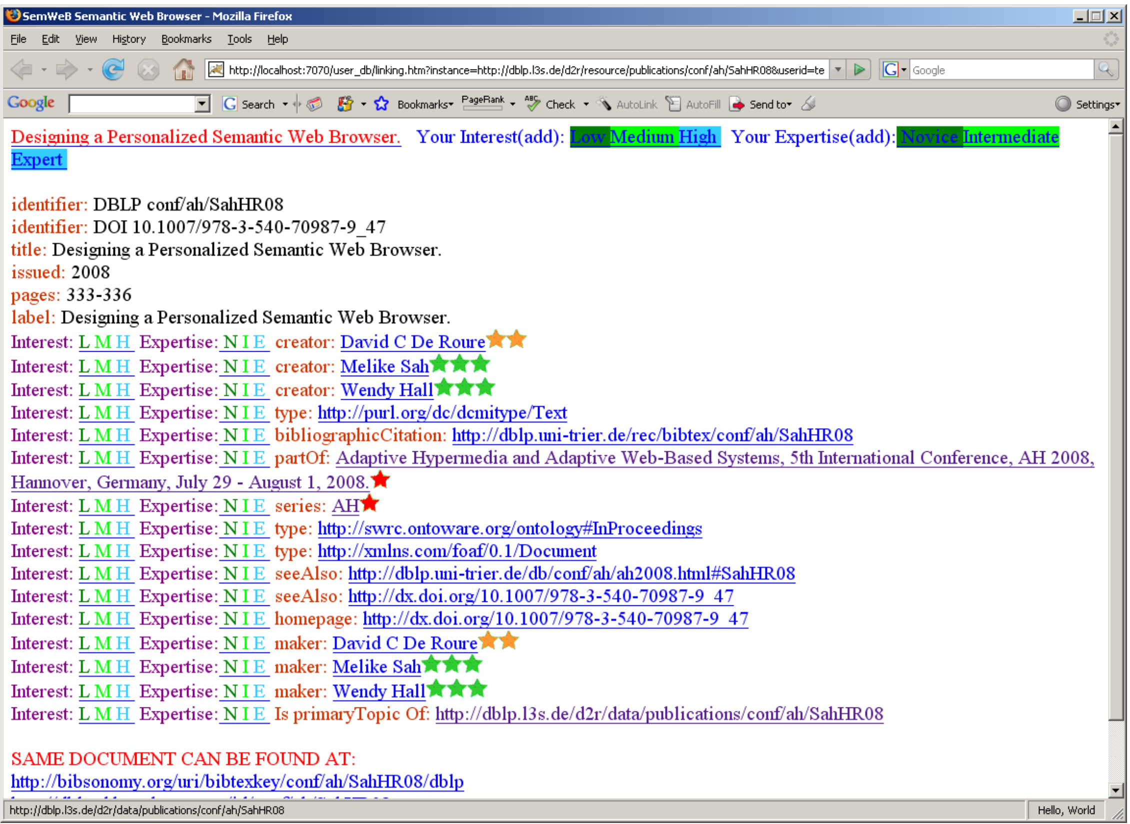


Figure 9-15 Annotated links with visual cues in the DBLP domain

9.1.1.3 Using Any Dereferenceable URI for Creating Semantic Links in SemWeB

In SemWeB, most of the updates are needed for IE and the semantic annotation module. However, SemWeB can be used directly by accessing the SLS from the address bar of an ordinary Web browser, such as Internet Explorer or Firefox. In this way, personalized semantic links and information can be generated. For instance, users can access the SLS service by entering “http://localhost:7070/user_db/linking.htm” to the address bar of any browser together with the semantic instance URI (which the user want to learn more information about), userid and password (if the user wants personalization), and generic goal services names as shown in Figure 9-16). Then, ontology-based links are generated and presented independent of ontologies and URIs. In Figure 9-17, the user requested “http://data.semanticweb.org/person/tom-heath” FOAF linked data URI and “SRSIW=true” goal service directly from the address bar.

```
http://localhost:7070/user_db/linking.htm?instance=URI&userid=ID&password=PASS&SRSIW=fa
lse;DBD=false;FMLDB=false;
```

Figure 9-16 The URL request to the SemWeB Semantic Linking Servlet (SLS)

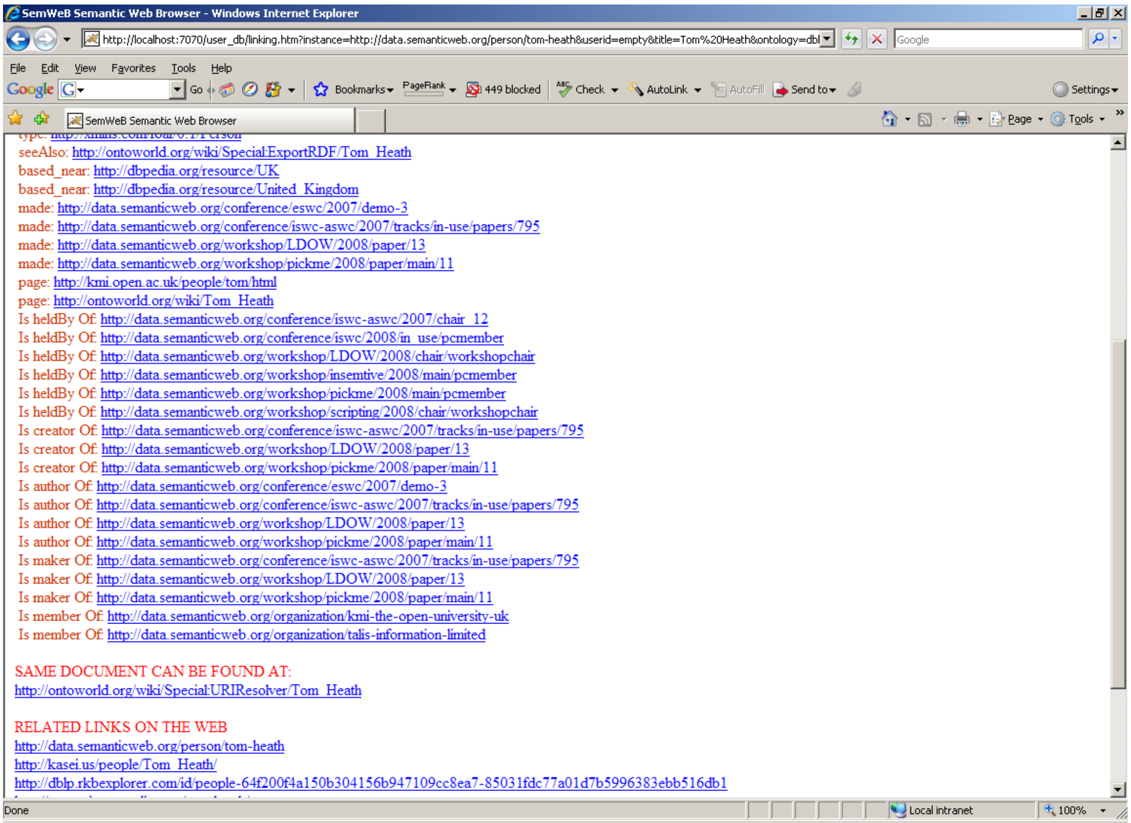


Figure 9-17 A URI is directly requested from the Internet Explorer using a valid URI

9.1.2 Adaptability

We have successfully experimented with SemWeB on the ECS, DBpedia and DBLP domains. It is evident that SemWeB is not an application-specific software and is adaptable with different linked data domains. The tests showed that the IE and annotation modules require modifications on different linked data domains, where we updated the modified GATE framework with new gazetteers, JAPEC rules and mapping database, also the sidebar is extended with the new ontological classes. However, the SLS and adaptation modules are generic and can be utilized on different linked data URIs. For instance, the SLS can make use of any linked data URI for semantic link creation. Also, the proposed user model can be applied to diverse ontologies and URIs and SemWeB generic goal services can be tested on diverse datasets. Furthermore, the proposed semantic relatedness measure can be applied to different datasets since it utilizes general ontological relationships (i.e. object properties, rdfs:seeAlso and owl:sameAs) for the calculation of the similarity value.

9.1.3 Scalability

This section discusses the scalability of the SemWeB system design and its modules.

9.1.3.1 Scalability of the SemWeB System Design

SemWeB is implemented as a browser extension. Thus, it only requires a JavaScript-enabled browser. All the functionalities of SemWeB are supported by the server-side, which means the client is lightweight in terms of memory and computation power. For the interaction between clients and the server, we adopted AJAX, which enables us to communicate with the clients asynchronously without interfering with the browsing of users.

9.1.3.2 Scalability of the Semantic Annotation

Scalability of GATE and the Mapping Database: Semantic annotation is very important for discovering semantic instances from Web pages and recommending new semantic links. While selecting an annotation framework, we choose a mature and adaptable software; GATE. Since, GATE is implemented as PR units, different parts of the semantic annotation can be improved or new modules can be added. JAPEC rules can be easily adapted to different ontology domains with minimum changes (i.e. gazetteer’s major or minor types were modified in JAPEC rules). In the IES module, we store URI-lexicon pairs a mapping database. Currently, we have utilized the MySQL database for storing mapping information.

Discussion of the Mapping Database: When the size of the mapping database increases, the querying time increases as well. In the ECS domain, the database was lightweight, thus query times were very reasonable. But, when we tested SemWeB on DBpedia, the mapping table has reached ~2.8 million rows and query times did not scale. To solve this problem, we indexed the mapping table according to the lexicon and URIs. Now, the query times are very reasonable; approximately 0.03 seconds for each query in DBpedia and approximately 0.01 seconds for each query in the ECS domain.

Scalability of Annotation Creation Times: Annotation creation times are important, if we want users to adopt our approach. Average Web users are often impatient and willing to see results as soon as possible. Therefore, annotations should be created in reasonable time without interfering with the browsing of the users. Firstly, we are not embedding links automatically; we show links if the user wants some guidance. In addition, the semantic annotation request is requested using AJAX and the results are sent back to the user asynchronously. Therefore, while the user is waiting, she can browse the Web page and the page does not freeze or browsing speed does not slow. When the semantic annotation is available, we show this information in the sidebar (i.e. "The page is successfully annotated" message is shown at the sidebar), which means users can use the sidebar to add more hyperlinks to the Web page.

To alleviate annotation creation times, we stored the created semantic annotations to the server-side. In this way, Web pages are collaboratively annotated by different users. If a Web page is annotated once, then the annotation is returned to the browser without overhead. In addition, we do not send duplicated semantic annotations to the browser. If an instance is recognized in more than one place, we only send it once and the sidebar is responsible for displaying the annotation in more than one place. In Tables 9-1 and 9-2, we show some example semantic annotation times using the ECS domain and DBpedia (i.e. average of 3 measurements). These annotation times were tested on a Windows XP computer, with 2.87 GB memory, 3.20 GHz CPU and 1 Gbps network connection. Note that we measured the time difference between the time the user requested the annotation and the time when the annotation was returned to the browser.

Table 9-1: Dynamic semantic annotation times using the ECS domain

Web Page	No of identified instances	Annotation time
http://www.ecs.soton.ac.uk/about/news/1440	32	4.776 secs
http://www.ecs.soton.ac.uk/people/nmg/publications	199	5.258 secs
http://www.ecs.soton.ac.uk/syllabus/COMP1002.html	37	3.135 secs
http://www.ecs.soton.ac.uk/about/news/2027	12	2.002 secs
http://www.ecs.soton.ac.uk/interests/java	19	2.995 secs

Table 9-2: Dynamic semantic annotation times using the DBpedia domain

Web Page	**No identified instances without** idf **filter**	**Annotation time without** idf **filter**	**No identified instances with filter (**$idf >= 3.0$**)**	**Annotation time with** idf **filter**
http://www.ecs.soton.ac.uk/about/news/2228	120	42.12 secs	20	60.716 secs
http://news.bbc.co.uk/1/hi/sci/tech/7829668.stm	147	42.1535 secs	72	91.469 secs
http://news.bbc.co.uk/1/hi/magazine/7818970.stm	212	44.432 secs	77	99.15 secs
http://www-history.mcs.st-andrews.ac.uk/history/HistTopics/The_Quantum_age_begins.html	229	45.187 secs	86	73.159 secs
http://www.nature.org/wherewework/northamerica/states/california/features/mammoth.html	190	43.8135	94	66.924 secs

As it is seen in Tables 9-1 and 9-2, the number of identified instances affects the duration of the annotation, since we query the mapping database to find the URI of each identified lexicon. In addition to the number of identified instances, the cache memory of the Web server affects the performance. In the ECS domain, we had small gazetteers (i.e. 900KB), however, in the DBpedia domain, the size of the gazetteers is approximately 60MB and the Web server cache memory is 1024 MB. Therefore dynamic annotations using the DBpedia domain takes longer than for the ECS domain because of the time needed to load the gazetteers. With a powerful Web server with high cache memory, we think that this problem can be alleviated because the gazetteers would be loaded faster. On the other hand, once a Web page has been annotated before, for future requests for the same page, annotation takes approximately 0.01 seconds to send to the browser from the server-side annotation storage module. In addition, in the DBpedia domain, annotation times slow down when link filtering is performed since we query Yahoo Search API over the HTTP protocol (see Table 9-2, last column). However, without link filtering, many DBpedia links are created.

9.1.3.3 Scalability of Semantic Link Creation

For link creation, we use dereferenceable URIs. The URI is dereferenced and links are created from the RDF description of the resource. Therefore, SemWeB can make use of any dereferenceable URI for semantic link creation and our system architecture can easily scale with different linked data domains. Generic goal services can also be applied to different domains. On the other hand, SemWeB can embed hyperlinks in any HTML document and is an open corpus hypermedia system.

9.1.3.4 Scalability of User Profiles

Architectural Scalability of User Profiles: Currently user profiles are stored in a central triple store at the server-side. For this purpose, we used the Jena triple store. There are advantages and disadvantages of storing user profiles to the server-side.

Advantages of using a central user database:

- When users login to personalization from different computers, the data can be reached from the central database. In the case of profiles stored in the individual's computer, the user lost the data when she logged in from another computer.
- When all user data is kept in one place, collaborative personalization can be provided. For instance, Amazon stores purchase of individual users and page views in a central database and uses these patterns to recommend products to other users.

Disadvantages of using a central user database:

- Querying user profiles from a database takes longer compared to querying from a local file. To solve this problem, before performing adaptation we first cache the user profile to a temporary Jena model and execute searches on the cached file.
- Scalability of the database is very important. Currently, we use the Jena triple store and the current triple storage limit is 20 million triples. In the future work, the Jena triple store will be upgraded with JXT (Garlik triple store) (Garlik, 2009). JXT can scale to 5-10 Gtriples (W3, 2009). This update of the architecture should support sufficient users for our system.

Accuracy and Completeness of User Profiles: In addition to architectural scalability, the accuracy and the completeness of the user profiles are important. For example, our user modelling is currently user-driven. If users are not willing to add concepts to their profiles then part of the adaptation cannot work. In future work, we are planning to automate user modelling, so that the user's browsing trails will be recognized and profiles will be updated automatically. Additionally, users will be able to add concepts explicitly into their profiles using the current functionalities.

9.1.3.5 Scalability of Personalization

AH is provided to personalize information to the needs of each individual user. Adaptation requires extra processing. In SemWeB new adaptive links and texts are created using browsing goals, also the relatedness between the user profiles and the created semantic links is measured to annotate links with visual cues. In order to provide adaptive links and contents efficiently, we need to keep time overhead as minimum as possible. To experiment the time overhead, we have performed some tests.

Has Interest:	http://id.ecs.soton.ac.uk/interest/ontologies Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/information_retrieval Value: Medium
Has Interest:	http://id.ecs.soton.ac.uk/interest/java Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/owl Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/logic_programming Value: Medium
Has Interest:	http://id.ecs.soton.ac.uk/interest/semantic_annotation Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/semantic_web Value: High
Has Interest:	http://id.ecs.soton.ac.uk/project/eprints Value: Low
Has Interest:	http://id.ecs.soton.ac.uk/interest/hypertext Value: Medium
Has Interest:	http://id.ecs.soton.ac.uk/project/ARGUS Value: Low
Has Interest:	http://id.ecs.soton.ac.uk/interest/scripting_languages Value: Low
Has Interest:	http://id.ecs.soton.ac.uk/interest/music Value: High
Has Interest:	http://id.ecs.soton.ac.uk/interest/web_programming Value: High
Has Interest:	http://id.ecs.soton.ac.uk/person/9677 Value: High
Has Interest:	http://id.ecs.soton.ac.uk/module/COMP1003/2007-2008 Value: Low
Has Interest:	http://id.ecs.soton.ac.uk/module/COMP1004/2007-2008 Value: Low
Has Interest:	http://id.ecs.soton.ac.uk/publication/13715 Value: High
Has Interest:	http://id.ecs.soton.ac.uk/presentation/290 Value: High
Has Interest:	http://dbpedia.org/resource/Wallace_Wattles Value: High
Has Interest:	http://dbpedia.org/resource/Category:New_Thought_writers Value: Higl
Has Interest:	http://dbpedia.org/resource/Category:Self-help_books Value: High
Has Interest:	http://dbpedia.org/resource/The_Science_of_Getting_Rich Value: High
Has Interest:	http://dbpedia.org/resource/The_Secret_(2006_film) Value: High

Figure 9-18 An example user profile for link annotation time measurement

Link Creation Times with and without Link Annotation: We tested the time overhead, when the link annotation was performed. During the experiment, in the user profile, there were 24 interests as shown in Figure 9-18. We selected some example semantic instances as shown in Table 9-3, then we measured the time needed to create semantic links with and without Link Annotation With Visual Cues (LAWVC). For each semantic instance, link creation times were measured 5 times. Then we took the average of them. These measurements were tested on a Windows XP operating computer, with 2.87 GB memory, 3.20 GHz CPU and 1 Gbps network connection.

Table 9-3: Link creation times with and without LAWVC using the ECS URIs

URI	No of RDF links created	Without LAWVC (mean)	With LAWVC (mean)	Adapted Links (coloured stars)
interest/semantic_web	48	0.2944 sec	9.205 sec	6 red, 1 orange and 1 green
interest/semantic _annotation	7	0.1504 sec	1.655 sec	3 red and 1 green
person/9677	15	0.1608 sec	2.872 sec	2 orange and 8 green
person/1650	52	0.2706 sec	6.5294 sec	1 green and 1 orange
person/5113	407	2.5696 sec	49.7355 sec	2 green stars
publication/13715	2	0.086 sec	0.8374 sec	1 green star

Table 9-4: Link creation times with and without LAWVC using the DBpedia URIs

URI	No of RDF links created	Without LAWVC (mean)	With LAWVC (mean)	Adapted Links (coloured stars)
dbpedia:Wallace_Wattles	17	4.6506 sec	11.9948 sec	1 red and 1 green
dbpedia:Category:New_T hought_Writers	48	1.3984 sec	28.123 sec	2 red and 2 green
dbpedia:Category:Spiritual _writers	30	1.1576 sec	16.463 sec	1 red, 1 orange and 1 green
dbpedia:Category:Spiritual _books	37	0.621 sec	18.7996 sec	4 green
dbpedia:Category:Self-help_books	104	2.5696 sec	58.301 sec	1 red and 1 green

As illustrated in Table 9-3, link annotation in the ECS domain is performed within a reasonable time when the number of created semantic links is low. The number of user interests and the number of RDF links in an RDF description affects the link creation times, since we need to dereference each URI at the run-time and compare similarity between them. Therefore, when the number of RDF links is increasing, link annotation time is also increasing. On the other hand, as shown in Table 9-4, link annotation time in the DBpedia domain takes longer since the network traffic at the DBpedia server during resolving a URI. We came to a conclusion that in a distributed KB system like linked data Web, caching is required to improve the performance of LAWVC.

Link Creation Times With and Without Goal Services: Since SemWeB uses distributed KBs in the goal services, the time needed to create links depends on the Web traffic at the requested server and the network bandwidth (i.e. 1 Gbps in the experiments). Based on some tests, we reported the delay added by each goal service as follows (note that these times can be slower or faster depending on the network traffic and Web server traffic): DBD and FMLDB goal services add approximately 0.1-0.3 seconds delay during the semantic link creation depending on the Web traffic at the DBpedia server. Link creation times with and without the PWW goal service depend on the number of projects with which the person is involved, the number of members of the project and the URI resolving time. In average, it adds 0.2-0.4 seconds delay. Again link creation times with and without the SRSIW goal service depends on the speed of the Sindice search engine. It adds approximately 0.2-0.4 seconds. In general, all services are executed within a reasonable time delay during link generation.

9.1.4 Discussion of SemWeB

9.1.4.1 Advantages

SemWeB is an open corpus hypermedia system that can embed ontology-based hyperlinks in HTML documents using the linked data. It also personalizes the information and links according to user profiles. Using linked data during ontology-based link creation and adaptation provides benefits to our system. The first benefit is *reusing*. Instead of creating new ontologies and populating them with metadata, existing resources are reused. This decreases the effort and time for creating our own ontology and metadata. In addition, since the data is hosted and looked after by other people, it does not have any *storage costs*. Second, in the real-world scenarios, data may come from many datasets located from diverse locations. Therefore testing and using linked data in Web browsing is *more realistic* than using specific metadata developed for a certain use-cases. For instance, many linked datasets provide up-to-date information (i.e. DBpedia) and we are not worried about the most difficult part, which is maintenance. Third, linked data provides solutions to achieve *open-corpus AH*. Open-corpus AH is very challenging, since the documents and their relations are not known at the design-time. In addition, user models should be related to a new set of information and able to cope with data coming from diverse domains. However these problems can be solved by linked data. In SemWeB, any Web page content can be annotated with metadata using linked data. Subsequently, linked data URIs are related to the user profiles. In this way, user profiles can dynamically expand and related to diverse datasets and we can achieve open-corpus AH.

9.1.4.2 Limitations

Although linked data supplies many benefits, there are things to be discussed, such as the correctness, completeness and quality of the metadata. The correctness of the metadata is one of the important aspects, for instance in DBpedia Wendy Hall also appears to be a tennis player (incorrect links). Therefore when relying on linked data, developers should also aware of this problem and may need to investigate the data. The other two problems that might limitate link creation are incomplete data and poor quality data (i.e. when very few RDF links and information about a resource is available). It should also be noted that the provided ontology-based links and goal services depend on the quality of the metadata. The final problem is changing and developing linked data. Currently, we manually analyze linked data and extract lexicons and URIs. In order to cope with new URIs and changing data, a Web crawler can be used to automatically extract information from the linked data Web.

Our semantic annotation is based on GATE. In GATE, documents are annotated using the lexicons provided in gazetteers, such that exact matches of the lexicons are searched in documents. However, in our approach, we do not solve co-reference problems. Co-reference is used to resolve occurrences of multiple identifiers for a single resource (Glaser et al., 2007). In our context, co-reference is the ability of recognizing variants of lexicons to the same reference (i.e. Nicholas Jennings or Nick Jennings represent same person). In our approach, if a lexicon is slightly different from the token in the text, then annotation fails. For instance, although “Nicholas Jennings” and “Nick Jennings” represent the same person, the semantic annotation module cannot recognize this instance using co-reference and dynamic linking fails. The other problem during semantic annotation is ambigious words. For instance, a lexicon might have more than one meaning, such as in DBpedia, Pascal represents a person, also an organization. As a result, incorrect links to Web resources might be generated.

Currently, in the DBpedia, we rely on the provided disambiguation links to solve this problem but in other linked data domains, we do not solve this problem. On the other hand accuracy of the semantic annotation can be improved in different ways. First, on different class instances (i.e. Person, Places), we can write heuristic rules to find variants of lexicons to solve co-reference and those variants can be added to the gazetteers. The second solution is to allow users to annotate instances manually from their browsers; they can highlight the instance names and provide a linked data URI to it. The improvements of the accuracy of the semantic annotations will be performed in the future work.

On the other hand, in SemWeB link creation is performed dynamically by dereferencing URIs at run-time. To improve the performance of link creation and link personalization, metadata caching is required and will be implemented in future. In addition, in the proposed semantic relatedness measure, we define different weights between resources. In future, these weights can be examined and different weight schemes can be applied in different ontology domains.

9.2 Scenario-Based Evaluation of SemWeB

This section explains use cases where SemWeB can provide benefits to the user, also we explain two possible scenarios in which SemWeB can be utilized to help users to locate related information using ontology-based links and AH.

9.2.1 Use Cases Where SemWeB Can Be Useful

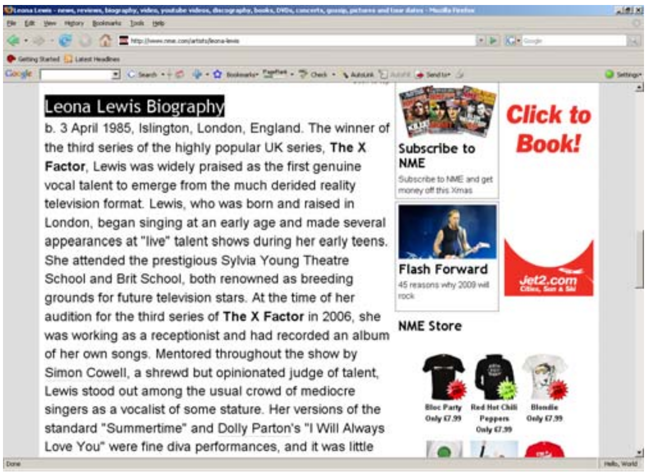
SemWeB is designed to support the browsing of users. Usually, users use search engines for finding Web resources but this is only half of the story. When users follow a link from search results, they have to read and understand page content and in general they are not guided during browsing. On the other hand, browsing is a complex activity and different browsing behaviours exist as discussed by (Bawden, 1986) and (Cove and Walsh, 1988) and explained in section 8.5.1. In SemWeB, our aim is to guide users to useful information during browsing. Browsing can be simply explained as clicking on hyperlinks and following from one Web page to another. Hyperlinks are the first-order objects within hypermedia. They allow us to navigate hyperspace and discover more information. However, there are limitations of links. Embedded links within the Web page can be insufficient for navigation, since links can be expensive to create and maintain. This results in loosely created links between Web resources. In cases, where Web pages have loosely created links, SemWeB can be employed to add ontology-based links to the Web page. Then, users can explore more information by navigation on the ontology-based hyperlinks. Thus, SemWeB can be utilized to overcome the insufficient link problem using ontology-based hyperlinks.

On the other hand, when too many links are provided to the user, users have difficulty in choosing the best links from a set of links, which can be facilitated by AH. AH aims to decrease this overhead by personalizing information and links according to the needs of the users (i.e. goal, preferences and interests). The second aim of SemWeB is to personalize information to individual users using ontology-based user models on the open-corpus Web. For this purpose, we developed a behaviour-based user model. For instance, if the user has browsing goals, then adaptive text and links are generated to guide users to related resources. Based on browsing interests, links are annotated with different visual cues and based on expertise, adaptive links and texts are created.

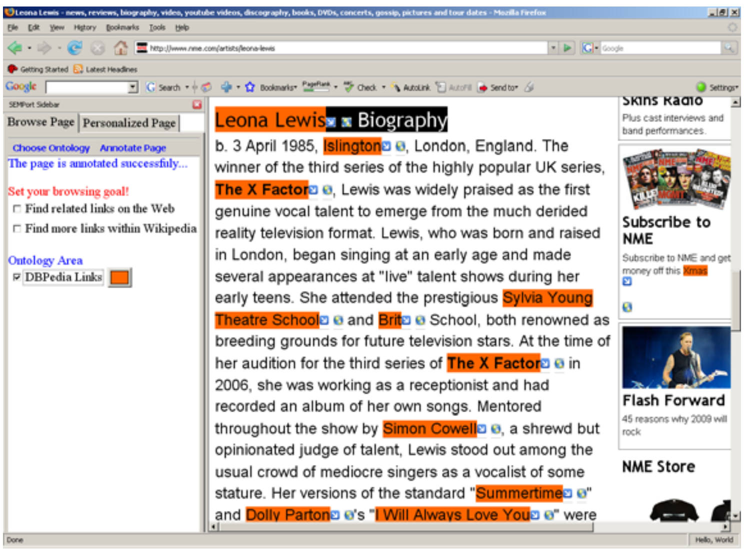
9.2.2 A Scenario using the DBpedia

Assume Ann was surfing on the Web looking for information about her favourite singer, Leona Lewis. She first made a Google search about her biography and came across an interesting Web page. However she was not able to find more links from this Web page to relevant pages (as shown in Figure 9-19 (a)). She decided to use SemWeB to find more links within the page context. She opened the SemWeB sidebar and logged in to SemWeB for personalized contents. First, she chose the DBpedia ontology, since she often reads articles from the Wikipedia. Then, she annotated the Web page and used the sidebar to add more links according to the context of the page as shown in Figure 9-19 (b). For instance, after the annotation, DBpedia links to Leona Lewis, recent album and some of her songs were added by SemWeB. After the annotation, Ann decided to use the ontology-based hyperlinks added by SemWeB. For instance, she wanted to learn more information about the singer and selected the “SRSIW” goal service from the sidebar and requested semantic links and information about “Leona Lewis”. In a new window, information about her DBpedia biography and links to her albums were shown together with related links created by the SRSIW goal service (Figure 9-19 (c)). The SRSIW goal service supplied relevant links to Leona’s list of songs, albums and her appearance on the X Factor, etc. From these links she followed through to some of the songs, X Factor and added some interests to her profile. For example, she added “Bleeding Love” song, “Pop Music” and “Contemporary R&B” categories and “Leona Lewis” to her interests (Figure 9-19 (d)). Then, she returned back to the originally annotated Web page. She clicked on the explore icon of the song “Bleeding Love” and in a new window, semantic links and more information are presented to her with the personalized hyperlinks as shown in Figure 9-19 (e). She clicked through some of the recommended links. For example, “Footprints in the Sand” is another Leona Lewis song recommended.

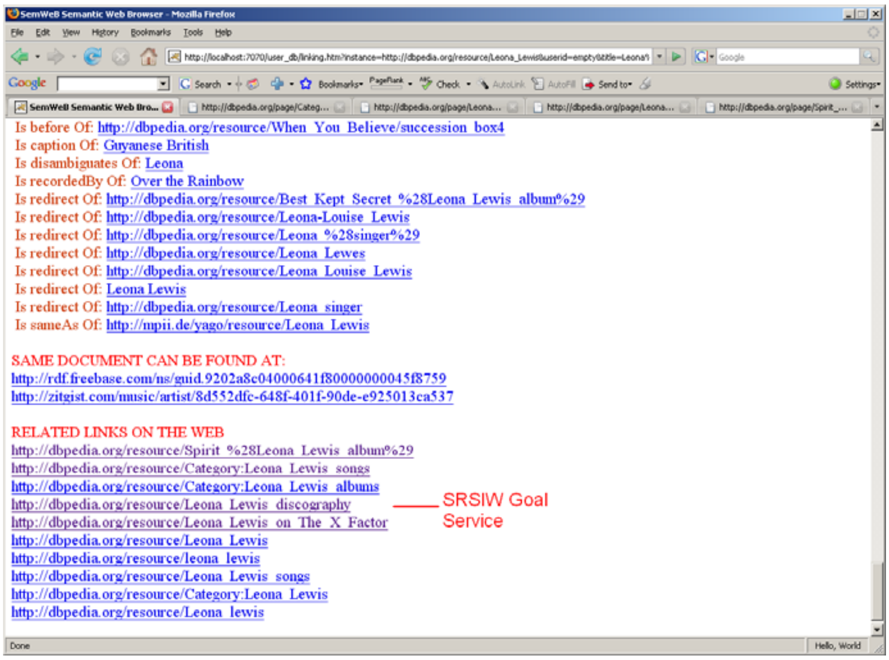
In this scenario, ontology-based hyperlinks added by SemWeB guided Ann to relevant resources and SemWeB helped to solve the insufficient link problem. Also, the goal service and recommended personalized links guided Ann to related Web resources.



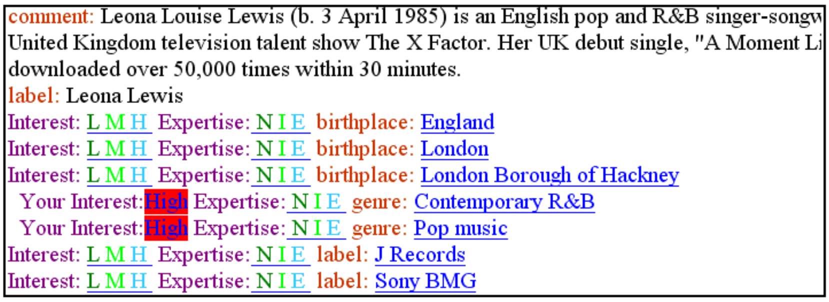
(a) Web page that does not have sufficient links to related Web resources



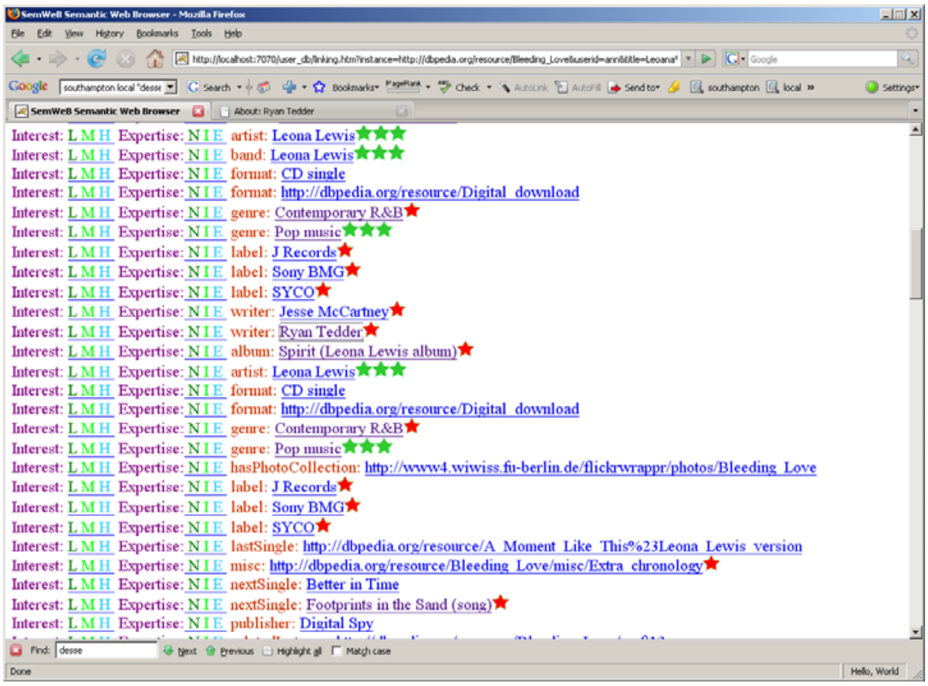
(b) The same page with the added semantic hyperlinks by SemWeB



(c) Ontology-based hyperlinks and goal-based adaptive links created by SemWeB



(d) Ann added some interests to her profile



(e) Recommended adaptive links by SemWeB

Figure 9-19 Demonstration of a scenario-based evaluation in the DBpedia domain

9.2.3 A Scenario using the ECS Domain

Jack is a postdoctoral researcher in ECS and was browsing news from the ECS website. He came across some interesting news but he couldn't find any links to related Web resources as shown in Figure 9-20 (a). He decided to use SemWeB and logged in to personalization as well. He annotated the Web page with the ECS ontology and highlighted people names, project names and ECS interests on the Web page (see Figure 9-20 (b)). He wanted to learn more about people involved in the project and he selected the "PWW" and "FRP" goal services. Then, he clicked on explore icons of people recognized on the Web page, such as "Alex Rogers" shown in Figure 9-20 (c). In the new Web page, he saw a tag of people related to this person and other projects related as well. He found these links interesting and followed to the "Robocop Rescue" project and added it to his profile. Then, he went to the ECS Web pages of related people, for instance to Nicholas Jennings's and Sarvapali Ramchurn's Web pages from the links provided by the PWW goal service. He also annotated these Web pages, and highlighted semantic instances of interests. Then, he requested the "agent-based computing" interest from SemWeB. As shown in Figure 9-20 (d), some of the links were annotated with visual cues, such as "Alex Rogers", "Sarvapali Ramchurn" and "Rajdeep Dash", since these people were members of the "Robocup Rescue" project. He followed through to Rajdeep Raj's Web page and found related publications.

In this scenario, we showed how SemWeB can guide users to useful links using ontology-based links. For instance, the insufficient link problem was solved by adding ontology-based links to the Web page by SemWeB. On the other hand, SemWeB goal services can provide useful related information to users according to their context. In this scenario, the user is provided with related links according to his goals. Moreover, using the interests of the user, the user is directed to relevant Web pages by using link annotation with visual cues.

(a) Web page that does not have sufficient links to related Web resources

(b) The same page with the added semantic hyperlinks by SemWeB

(c) Visualization of ontology-based and adaptive links

(d) User is guided to relevant Web resources using link annotation

Figure 9-20 Demonstration of a scenario-based evaluation in the ECS domain

9.3 Summary Evaluations

In the system-based evaluations, it has been shown that SemWeB is not domain-specific and can be easily tested on different linked data domains, such as the DBpedia and the DBLP domains. We showed how users can be guided to relevant resources on the Web using linked data and how this data can be personalized using ontology-based user models. Experiments showed that SemWeB can be successfully utilized to create open-corpus ontology-based hyperlinks and AH on Web documents using diverse linked data domains. With the use of linked data, metadata can be located by resolving linked data URIs. As a result, SemWeB is scalable with any linked dataset for ontology-based links creation and it can also adapt this information to individual users using ontology-based user models. We also discussed and tested the scalability of the SemWeB system design. Since, SemWeB uses distributed KBs to create and adapt data and information, the performance also depends on retrieving data from remote servers. To improve the performance of the system, caching is required. Currently, we cache semantic annotations on the server-side. In future work, resolved URIs can be cached to a local server to improve link creation times during the adaptation of links.

In the scenario-based evaluations, we have shown how SemWeB can be used to overcome the insufficient link problem with ontology-based hyperlinks and how users can be guided to relevant Web resources using AH. We showed that an ordinary Web browsing experience can be enriched with semantic and adaptive links and contents. In the next chapter, we discuss the overall conclusions of the thesis and discuss further research directions.

10 Conclusions and Future Work

This thesis has presented a novel personalized semantic portal and a novel personalized Semantic Web browser for context-based hyperlink generation and personalization using Semantic Web technologies. In this chapter, first we discuss our research objectives and findings, also present overall conclusions of SEMPort and SemWeB. Then, follow-up work and further research directions are given.

10.1 Research Objectives and Findings

Our research has been motivated by hypermedia systems, Semantic Web technologies and Adaptive Hypermedia (AH). In a hypermedia system, users can be guided to related contents by creating and presenting context-based hyperlinks. In addition, different users have diverse information needs and Adaptive Hypermedia (AH) is an alternative to the traditional "one-size-fits-all" static hypermedia systems. In our research, we developed hypertext systems using Semantic Web technologies to provide context-based hyperlinks using ontologies and ontology-based metadata, also to support personalization using ontology-based user models.

In this research, the researcher has shown how she successfully managed to support context-based hyperlink generation and personalization in a semantic portal and in a Semantic Web browser using Semantic Web technologies and ontology-based metadata. Our research findings can be summarized as follows:

- It is observed that Semantic Web technologies (i.e. ontologies) provided a flexible mechanism for sharing data.
- It is successfully shown that ontologies can be used to create rich ontology-based hyperlinks between Web resources by using inferencing.

- It is successfully demonstrated that ontology-based user models represented with semantic metadata (i.e. RDF) are interoperable within different domain ontologies.
- Linked data is a new trend of open source metadata and it is successfully illustrated that it can be utilized to create context-based hyperlinks within Web documents.
- It is successfully shown that linked data can be used to annotate Web documents with rich semantic metadata.
- It is observed and demonstrated that linked data provides new ways of relating user models to diverse Web resources on the Web by using resolvable linked data URIs.
- Analysis of user modelling standards showed that there is a need for new user models for accomplishing Web-based personalization, such as user models that represent browsing goals, interests and preferences. A new user model is proposed and successfully applied for Web-based personalization.

In the thesis, we have two different contributions; to semantic portals and Semantic Web browsers and we explain the overall conclusions within two sub-sections.

10.2 Conclusions of SEMPort

We introduced and presented a novel semantic portal, SEMPort, for improving linking between resources and providing AH in a semantic portal. We summarise our research findings and overall conclusions of SEMPort as follows:

- Ontologies provided a flexible mechanism for sharing data between portal users and a number of maintenance mechanisms are provided using ontologies. For instance, Protégé ontology editor can be used to update ontology/instances, a Web front-end can be used to upload ontology/instance files, and an easy-to-use distributed Web interface can be used for the edition and provision of the instances in real-time.
- To facilitate information access by using links to relevant and interesting concepts, we provided context-based semantic hyperlinks using ontologies and ontology-based metadata. For this purpose, explicit, reverse, implicit, and recommendation links are generated. We also applied reasoning to infer implicit relationships between different Web resources and present this data during semantic navigation.
- Ontology-based search is integrated to the semantic navigation, to perform concept-specific searches during the browsing.

- Personalization is supplied to adapt to the different needs of the users using ontology-based user models. Different AH techniques are designed to support the personalization: personalized homepages, link sorting and hyperlinks with visual cues. In addition, users can control their profiles from the SEMPort's Web interface and can contribute to the metadata by adding semantic bookmarks. With the use of ontology-based user models, interoperability between the user models and the domain model concepts is provided easily.
- SEMPort is implemented with re-usable components enabling different domains to be tested with a low-cost. For instance, semantic navigation and ontology-based search are generic and can be adapted to different domains. Semantic hyperlinks can be easily adapted to diverse domains, by simply changing the rules. However, personalization is needed to be adapted to the portal domain.
- For the evaluation and illustration of our approach, we tested SEMPort on the ECS CMWP and carried out user studies on the ECS CMWP and SEMPort using a set of tasks. The results of the tests showed that users performed tasks better using SEMPort (98% correct answers using SEMPort and 41% correct answers using the ECS CMWP). Besides, questionnaires were used to measure general opinion of users about the system and its usability. The results of the questionnaire showed that most users enjoyed using SEMPort and 100% of the subjects preferred to continue to use it. Personalization and links were the most preferred features of the portal and subjects scored 4.0 or higher (out of 5) for the satisfaction with the all of the functionalities of SEMPort: Semantic navigation (4.4), semantic hyperlinks (4.5), personalized homepages (4.4), link sorting (4.0), link with visual cues (4.1), content editing/provision (4.4).
- To determine any usability problems of SEMPort, structured review was conducted using Nielsen usability heuristics. A number of usability problems were identified by these reviewers and most of the problems identified were fixed before the user testing. The use of Nielsen heuristics formed a sound basis for this evaluation.

10.2.1 Summary of Caveats from Evaluations

In our opinion, we could have improved the evaluations undertaken with more in-depth analysis:

- A pre-study could have helped to understand needs of ECS CMWP users.

- In the empirical study, undergraduate students could have been used instead of postgraduate students, since undergraduates use ECS CMWP for selecting and following modules during the semester.
- In the structured review, five evaluators could have been used instead of four and three iterative evaluations could have been done to identify more usability problems of SEMPort.

10.3 Conclusions of SemWeB

We introduced and presented a personalized Semantic Web browser, SemWeB, for generating context-based hyperlinks and AH using linked data within Web documents. SemWeB is an extension to the Mozilla Firefox Web browser. It extends the browser with a sidebar. Where, users can use this sidebar to gain access to ontology-based hyperlinks and personalized information. We summarise our research findings and overall conclusions of SemWeB as follows:

- In SemWeB, users are not required to adopt the wholesale vision of the Semantic Web, but they seamlessly supported by semantic links based on dynamically annotated Web pages using a linked data domain. The SemWeB sidebar can be used to highlight ontological instances, which also embeds links to Web pages. Using the embedded links, users can request semantic links and personalized information. SemWeB is an open-corpus system such that dynamic semantic hyperlinks and AH can be created on different Web domains.
- We demonstrated that Web pages can be annotated using a linked data domain. For this purpose, we utilized GATE framework. We extended GATE with a lookup service and annotation generation and storage modules. Our experiments using the ECS, DBpedia and DBLP domains showed that SemWeB is successfully utilized to annotate Web pages using a linked data domain.
- To provide relevant information and hyperlinks to users, Semantic Linking Servlet (SLS) and goal-based services are implemented. Semantic hyperlinks are generated by dereferencing linked data URIs and extracting RDF links. In addition, we developed diverse goal services to show implicit and related information and links to users from distributed linked data resources on the Web. In our experiments, we observed that resources from linked data can be easily combined using vocabularies

and formal semantics (RDF). In this way, unified information can be presented to users by searching and combining data from distributed linked data resources.

- Analysis of the existing user modelling standards showed that there is a need for new user models which can support user's browsing as well. For supporting AH during Web browsing, we developed a new behaviour-based and ontology-driven user model. In this model, we introduced new concepts, such as *browsing goal, interest, expertise* and *browsing behaviour*. Users can explicitly update their profiles, such as they can add automatically provided browsing goals, or indicate their interests or expertise to semantic instances from their browsers. SemWeB also implicitly updates browsing behaviours.
- With SemWeB, users can add interests, expertise and goals to any resolvable linked data URI on the Web. In this way, user profiles can be extendable with diverse data without the control of our system, which allow us to achieve open-corpus AH. Our experiments with the ECS, DBpedia and DBLP illustrated that the proposed user model is successfully applied to different domains.
- To adapt information and links to the needs of the user in open Web environment, we provided different AH methods and techniques using the proposed user model: adaptive link and text generation based on browsing goals, link annotation with visual cues based on a proposed a new semantic relatedness measure, expertise based link and text generation and personalized homepages.
- SemWeB relies on open standard linked data for the semantic annotation, ontology-based link generation and personalization. Therefore, SemWeB is scalable with any linked data domain. The experiments with the ECS, DBpedia and DBLP domains showed that SemWeB is scalable, adaptable and interoperable with different linked data domains. The experiments also showed that IE and semantic annotation unit requires some modifications, however, SLS and adaptation modules are generic and works on different URIs and ontologies.
- In the scenario-based evaluations, we illustrated benefits of SemWeB in real use case scenarios. For instance, SemWeB is useful when the Web page has insufficient hyperlinks and users can use SemWeB to discover semantic hyperlinks to the relevant Web pages. In addition, with SemWeB, presented information and links are adapted to the browsing needs of users using AH methods and techniques.

10.3.1 Limitations of SemWeB

SemWeB is intended to achieve dynamic semantic linking and AH in open corpus Web and our contributions are presented in the previous section. However, there are limitations of our approach which needs to be solved, such as:

- Ontology selection within the browser is manual. We assume that the user will choose an ontology for use with the browser.
- Dynamic semantic annotation requires extra time and the scalability of semantic annotation needs further investigation (i.e. improving annotation times and resolving co-reference problems).
- We rely on linked data for semantic link creation instead of generating our own metadata. However, quality, correctness and completeness of the linked data may limitate our link generation approach.
- Dynamic link annotation with visual cues using open linked data requires extra time for dereferencing metadata. In a distributed knowledge base like linked data Web, caching of linked data is necessary to improve the performance of link annotation.

10.4 Future Work

10.4.1 System Enhancements

Some issues in the developed systems require further investigation. The potential work is outlined as follows:

Short-Term Plans:

SEMPort: In SEMPort, currently users cannot add new instances from the content editing and provision Web interface. In a developing and expanding information space in a portal, new information should be able to be added and in future, this functionality will be added.

SemWeB: Currently user modelling in SemWeB is user-driven. In future work, this process will be automated. For example, the interests of the users can be understood from the browsing trails and those interests can be automatically added to the user profiles. In addition, a goal search box can be added to the SemWeB sidebar, as such browsing goals can be dynamically obtained from users. We also have plans to provide direct-guidance, such that the best links can be shown to the user at the SemWeB sidebar. This can be achieved during the semantic annotation, as such, created annotations can be personalized and the best links can be presented to the user according to their profiles (e.g. using browsing interests or browsing goals). We will also upgrade GATE gazetteers to improve semantic annotation times. For example, we can create database gazetteers to speed up the annotation. In future work, improvement of the accuracy of the semantic annotations will be considered as well, such as new annotations can be provided by the user from the browser. We have plans to improve the proposed semantic relatedness measure. For instance, hierarchical similarties between entities can be amalgated to the current semantic relatedness measurement, which can be calculated by using Wordnet or YAGO[64] categories. In addition, in different linked data domains, the edge weights defined between entities can be modified according to important relationships between entities. The HCI aspects of SemWeB will be also improved. For instance, instead of showing hyperlinks in a new Web page, semantic hyperlinks can be shown in a pop-up window. Finally, we will perform a user-based evaluation on SemWeB to assess the quality of the created semantic hyperlinks and adaptive contents from the user's perspective. In this user study, we will also evaluate which user characteristics (expertise, browsing goals and browsing interests) are more useful for Web-based personalization.

Long-Term Plans:

SemWeB: In SemWeB, different linked data domains will be tested and experimented on users. Since the quality of the metadata affects the created hyperlinks, an overall comparison of the linked data domains can be done and tested on users. In addition, we will work on security and trust aspects. For instance, users can sign up for the trusted metadata in SemWeB. Also, we need to consider the security of the user profiles.

10.4.2 Research Directions

We proposed and introduced a semantic portal for improving information sharing between portal members, enhancing linking between portal resources and providing AH using Semantic Web technologies. With our approach, we have showed that

[64] http://www.mpi-inf.mpg.de/yago-naga/yago/ [last accessed, 24/3/2009]

existing semantic portal approaches should consider the needs of the users as well as developers of the portal. For instance, users require links to related Web resources during browsing. Also portals have enormous amounts of information and it is beneficial of the user to personalize the contents to improve information discovery. In addition, editing and maintaining the contents of the portal is very important for the development cycle of a semantic portal. With SEMPort, we experimented and showed different functionalities for the state-of-the-art semantic portals, which both support portal's users as well as portal's developers. The researcher came to a conclusion that for a portal to be accepted by large masses of users, the needs of the users should be also considered.

To bring the Semantic Web metadata to the everyday Web browsing, we proposed and introduced a Semantic Web browser. In this approach, we showed that users can benefit from the linked data without wholesale adopt the vision of the Semantic Web; with SemWeB we found a way of transparently connecting data Web (linked data) into the document Web (WWW). In this way, ordinary Web users can benefit from semantic metadata. In addition, we proposed ideas for personalizing Web browsing, such as we introduced a behaviour-based and ontology-driven user model for Web-based personalization. With our approach, AH can be achieved in any Web domain, since users will use their Web browsers and personalization is provided during browsing. Linked data is a new trend of open source metadata and its popularity is growing over time (i.e. DBpedia, MusicBrianz, etc.). With such open standard metadata, browsing can be improved with automatically created enrich semantic hyperlinks. For instance, information from different linked data resources can be combined and presented, to help users to find related Web resources with the use of Semantic Web technologies. Such intelligent semantic linking can reduce information overhead of users and can improve information discovery. In addition, the Web has enormous amounts of information and it is difficult to locate right information. AH can alleviate this problem by tolerating data to the needs of users and such personalization in open Web environment can provide many benefits to users, such as reducing cognitive overhead.
In this context, SemWeB is one of the pioneer applications that uses linked data and it connects data Web to the document Web with the aim of better information discovery using semantic links and personalized contents. In future, the researcher thinks that SemWeB like applications will bring new ways of linking and personalization in

context in Web-scale hypermedia applications. In addition, SemWeB demonstrated new ways of using linked data within everyday Web browsing and may contribute to the adoption of linked data among users and developers.

References

Abel, F., and N. Henze (2005). *User Awareness and Personalization in Semantic Portals*. In Proceedings of the International Semantic Web Conference.

ACM CCS (2008). *The ACM Computer Classification System*. Retrieved December 19 2008, from www.acm.org/class/1998/ ccs98.txt.

Akscyn, R. M., D. L. McCracken and E. A. Yoder (1988). “KMS: A distributed hypermedia system for managing knowledge in organizations”. Communications of the ACM, 31(7), 820-835.

Anderson, K. M., R. N. Taylor and E. J. Whitehead (2000). “Chimera: Hypermedia for Heterogeneous Software Development Environments”. ACM Transactions on Information Systems, 18(3), 211-245.

Andrews, K., F. Kappe and H. Maurer (1995). “The Hyper-G network Information Systems”. The Journal of Universal Computer Science, 1(4).

Aroyo, L., P. De Bra, G. J. Houben and R. Vdovjak (2004). “Embedding information retrieval in adaptive hypermedia: IR meets AHA!”. New Review of Hypermedia and Multimedia, 10(1), 53–76.

Ashman, H. and J. Verbyla (1993). *The functional model of the link*. In Proceedings of Hypertext, Seattle, WA.

Ashman, H., A. Garrido and H. Oinas-Kukkonen (1997). *Hand-made and Computed Links, Precomputed and Dynamic Links*. In Proceedings of Hypermedia - Information Retrieval - Multimedia Conference, 191-208.

Auer, S., C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak and Z. Ives (2007). *DBpedia: A Nucleus for a Web of Open Data*. International Semantic Web Conference+Asian Semantic Web Conference, LNCS, 4825, 11–15.

Baeza-Yates, R. and B. Ribeiro-Neto (1999). Modern Information Retrieval. ACM Press, Addison-Wesley, New York.

Bailey, C., W. Hall, D. Millard and M. Weal (2002). *Towards open Adaptive Hypermedia*. In Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems.

Bajraktarevic, N., W. Hall and P. Fullick (2003). *Incorporating learning styles in hypermedia environment: Empirical evaluation*. In the Workshop on Adaptive Hypermedia and Adaptive Web-based Systems. Retrieved February 27 2009, from http://wwwis.win.tue.nl/ah2003/proceedings/paper4.pdf

Bawden, D. (1986). "Information Systems and the Stimulation of Creativity". Journal of Information Science, 12, 203-216.

Bechofer, S., L. Carr, C. Goble, S. Kampa and T. Miles-Board, T. (2002). *The Semantics of Semantic Annotation*. In Proceedings of First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, 1151-1167.

Berners-Lee, T. and M. Fischetti (1999). Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor, Harper, San Francisco.

Berners-Lee, T (1998). *Semantic Web Roadmap*. Retrieved December 19 2008, from http://www.w3.org/DesignIssues/Semantic.html.

Berners-Lee, T (2000).: *Semantic Web Stack*. Presentation at XML 2000, Retrieved December 19 2008, http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html.

Berners-Lee, T., J. Hendler, and O. Lassila (2001). "The Semantic Web". Scientific American, 284(5), 34-43.

Berners-Lee, T., W. Hall, J. A. Hendler, K. O'Hara, N. R. Shadbolt and D. J. Weitzner (2006). A Framework for Web Science. Foundations and Trends in Web Science, 1(1), 1-130.

Berners-Lee, T., Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer and D. Sheets (2006b). *Tabulator: Exploring and analyzing linked data on the semantic web*. In Proceedings of the 3rd International Semantic Web User Interaction Workshop.

Berners-Lee, T (2006c). *Linked Data*. Web architecture note. Retrieved December 19 2008, from http://www.w3.org/DesignIssues/LinkedData.html.

Bernstein, A., E. Kaufmann, C. Bürki and M. Klein (2005). *How Similart Is It? Towards Personalized Similarity Measures for Ontologies*. In Internationale Tagung Wirtschaftsinformatik, Book chapter, 1347-1366.

Bizer, C. and T. Gauss (2007). *Disco - hyperdata browser*. Retrieved December 19 2008, from http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/.

Bizer, C., R. Cyganiak and T. Heath (2007). *How to Publish Linked Data on the Web*. Retrieved December 19 2008, from http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/.

Bizer, C., R. Cyganiak and T. Gauss (2007b). *The RDF Book Mashup: From Web APIs to a Web of Data*. In the Workshop on Scripting for the Semantic Web.

Bizer, C. and R. Cyganiak (2007). *D2R Server: Publishing Relational Databases on the Semantic Web*. Retrieved December 19 2008, from http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/.

Brown, P.J. (1987). *Turning Ideas into Products: the Guide System*. In Proceedings of the ACM Hypertext Workshop. 33-40.

Brunkhorst, I., and N. Henze (2005). *User Awareness in Semantic Portals*. International Workshop on Personalization on the Semantic Web (PerSWeb'05).

Brusilovsky, P. (1996). "Methods and techniques of Adaptive Hypermedia". User Modeling and User-Adapted Interaction, 6(2-3), 87-129.

Brusilovsky, P. (2001). "Adaptive Hypermedia". User Modeling and User Adapted Interaction, 11, 87-110.

Brusilovsky, P. (2003). "Developing adaptive educational hypermedia systems: From design models to authoring tools". In: Murray, T., Blessing, S., and Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environment. Dordrecht: Kluwer Academic Publishers.

Brusilovsky, P. And R. Rizzo (2003). Using maps and landmarks for navigation between closed and open corpus hyperspace in Web-based education. The New Review of Hypermedia and Multimedia. Volume 8, 59-82.

Brusilovsky, P. (2004). "Adaptive Navigation Support: From AH to the Adaptive Web and Beyond". PsychNology Journal, 2(1), 7-23.

Brusilovsky, P., and N. Henze (2007). "Open corpus adaptive educational hypermedia". In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web, LNCS, 4321, 671– 696. Springer, Heidelberg.

Brusilovsky, P. (2008). *Adaptive Navigation Support for Open Corpus Hypermedia Systems*. Invited Talk in Adaptive Hypermedia and Adaptive Web-based Systems Conference, LNCS, 5149, 6-8.

Bush, V. (1945). "As We May Think". The Atlantic Monthly.

Cailliau R. (1995). *Short History of the Web*. Retrieved from 20 December 2008. from http://www.netvalley.com/archives/mirrors/robert_cailliau_speech.htm.

Carmel, E., S. Crawford and H. Chen (1992). "Browsing in Hypertext: A Cognitive Study". IEEE Transactional on Systems, Man, and Cybermetics, 22(5), 865-884.

Carmody, S., W. Gross, T. H. Nelson, D. Rice and A. Van Dam (1969). *A Hypertext Editing System for the 360*. In Proceedings of the Second University of Illinois Conference on Computer Graphics, 291-330.

Carmona, C., D. Bueno and R. Conejo (2002). *SIGUE: Making Web Courses Adaptive*. In Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, 376-379.

Carr, L. A., D. C. De Roure, W. Hall and G. J. Hill (1995). "The Distributed Link Service: A tool for Publishers, Authors and Readers". In World Wide Web Journal, 1(1).

Carr, L., W. Hall, S. Bechhofer and C. Goble (2001). *Conceptual Linking: Ontology-based Open Hypermedia*. In Proceedings of International World Wide Web Conference, Hong Kong, 334-342.

Castells, P., M. Fernandez, D. Vallet (2007). "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval". IEEE Transactions on Knowledge and Data Engineering, 19(2), 261-272.

Cimiano, P., S. Handshuh, S. Staab (2004). *Towards the self-annotating web.* In: Proceedings of the International World Wide Web Conference.

Conlan, O., V. Wade, C. Bruen and M. Gargan (2002). *Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning*. In International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, LNCS, 2347, 100-111.

Conlan, O., C. Hockemeyer, V. Wade, D. Albert, and M. Gargan (2002b). *An Architecture for integrating Adaptive Hypermedia Service with Open Learning Environments*. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications.

Corcho, O., A. Gomez-Perez, A. Lopez-Cima, V. Lopez-Garcia, and M. C. Suarez-Figueroa (2003). *ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets*. International Semantic Web Conference, 2870 802-817.

Corcho, O., A. Lopez-Cima, and A. Gomez-Perez (2006). *The ODESeW 2.0 Semantic Web Application Framework*. International World Wide Web Conference, Scotland.

Cove, J., and B. Walsh (1988). "Online Text Retrieval via Browsing". Information Processing and Management, 24, 31-37.

Cunningham, H., D. Maynard, V. Tablan (2000). *JAPE – a Java Annotation Patterns Engine*. Technical Report, University of Sheffield. Retrieved December 19, from http://citeseer.ist.psu.edu/article/cunningham00jape.html.

Cunningham, H., D. Maynard, K. Bontcheva and V. Tablan, V. (2002). *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In Proceedings of 40th Anniversary Meeting of the Association for Computational Linguistics.

Dagger, D., O. Conlan, and V. Wade (2003). *An Architecture for Candidacy in Adaptive eLearning Systems to Facilitate the Reuse of Learning Resources*. In Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, 49-56.

DBpedia (2008). *The DBpedia Knowledgebase*. Retrieved December 19 2008, from http://dbpedia.org/About.

DBLP (2008). *DBLP Bibliography Database.* Retrieved December 19 2008, from http://dblp.l3s.de/d2r/.

De Bra, P., P. Brusilovsky, and G-J. Houben (1999). "AH: From Systems to Framework". In the Journal of ACM Computing Surveys, 30(4).

De Bra, P., A. Aerts, D. Smits and N. Stash (2002). *AHA! Meets AHAM*. In Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS, 2347, 388-391.

De Bra, P., D. Smits, and N. Stash (2006). *Creating and Delivering Adaptive Courses with AHA!*. European Conference on Technology Enhanced Learning, Lecture Notes in Computer Science, 4227, 21-33.

Decker, S., M. Erdmann, D. Fensel and R. Studer (1999). *Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information*. Semantic Issues in Multimedia Systems. 351-369. Kluwer Academic Publishers.

Diederich, J, M., Dzbor and D. Maynard (2007). "REASE—The repository for learning units about the Semantic Web". New Review of Hypermedia and Multimedia, 13(2), January 2007 , 211 – 237.

Dill, S., N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin and J. Y. Zien (2003). *SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation*. In Proceedings International World Wide Web Conference.

Ding, Y. (2005). Study of Design Issues on an Automated Semantic Annotation System. AIS SIGSEMIS Bulletin, 2, 45-51.

DOAP (2008). Retrieved December 19 2008, from http://trac.usefulinc.com/doap.

Dolog, P. and W. Nejdl (2003). *Challenges and Benefits of the Semantic Web for User Modelling*. In Proceedings of Workshop on AH and Adaptive Web-Based Systems at WWW, UM and Hypertext Conference.

Dolog, P., N. Henze, W. Nejdl and M. Sintek (2003). *Towards an Adaptive Semantic Web*. In Proceedings of the Principles and Practice of Semantic Web Reasoning (PPSWR'03) conference, Mumbay, India.

Dolog, P., R. Gavriloaie, W. Nejdl and J. Brase (2003b). *Integrating AH techniques and open rdf-based environments*. In Proceedings of International World Wide Web Conference, Budapest, Hungary.

Domingue, J., M. Dzbor and E. Motta (2004). *Collaborative Semantic Web Browsing with Magpie*. In European Semantic Web Symposium, LNCS, Volume 3053.

Dublin Core (2006). *Dublin Core Metadata Element Set, Version 1.1*. Retrieved December 19 2008, from dublincore.org.

Dumbill, E. (2002). XML Watch: Finding friends with XML and RDF: The Friend-of-a-Friend vocabulary can make it easier to manage online communities. Retrieved December 19 2008, http://www.ibm.com/developerworks/xml/library/x-foaf.html.

Dzbor, M., J. Domingue and E. Motta (2003). *Magpie – towards a semantic web browser*. In Proceedings of the International Semantic Web Conference.

Dzbor M., E. Motta and J. Domingue (2007). "Magpie: Experiences in supporting Semantic Web browsing". In Journal of Web Semantics, 5(3), 204-222.

ECS CMWP (2006). *ECS Course Modules Web Page*. Available from https://secure.ecs.soton.ac.uk/notes/

ECS Ontology (2006). ECS Ontology. Retrieved December 19 2008, from rdf.ecs.soton.ac.uk/ontology/ecs.

ECS Southampton (2006). *ECS Southampton Ontology Information*, Retrieved December 19 2008, from http://id.ecs.soton.ac.uk/docs/

eduPerson (2007). Retrieved December 19 2008, from http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson-200712.html.

Ehrig M., S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G.Stumme, Y. Sure, J. Tane, R.

Volz, V. Zacharias (2002). *KAON - Towards a large scale Semantic Web*. In E-Commerce and Web Technologies International Conference, LNCS, 2455.

Ehrig, M., P. Haase, M. Hefke and N. Stojanovic (2004). *Similarity for ontologies - a comprehensive framework*. In Workshop of Enterprise Modelling and Ontology: Ingredients for Interoperability, International Conference on Practical Aspects of Knowledge Management.

Eirinaki, M. and M. Vazirgiannis (2003). "Web Mining for Web Personalization", ACM Transactions on Internet Technology, 3(1), 1-27.

Eklund, J., P. Brusilovsky and E. Schwarz (1997). *Adaptive Textbooks on the WWW*. In the Third Australian Conference on the World Wide Web, 186–192.

Engelbart, D.C. (1963). "Conceptual Framework for the Augmentation of Man's Intellect". In Vistas of Information Handling, (1), Spartan Books.

Farzan, R., M. Coyle, J. Freyne, P. Brusilovsky and B. Smyth (2007). *ASSIST: adaptive social support for information space traversal*. In the Proceedings of 18th conference on Hypertext and hypermedia, 199-208.

Fielding, R., J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee (1999). *Hypertext Transfer Protocol – HTTP 1.1*. Retrieved December 19 2008, from http://tools.ietf.org/html/rfc2616.

FOAF (2007). *FOAF Vocabulary Specification 0.9*. Retrieved December 19 2008, from http://xmlns.com/foaf/0.1/.

Fountain, A., W. Hall, I. Heath and D. C. Davis (1990). *MICROCOSM: An Open Model for Hypermedia with Dynamic Linking*. Hypertext: Concepts, Systems and Applications, Proceedings of the European Conference on Hypertext, 298-311.

Garlik (2009). *Garlik JXT Triple Store*. Retrieved January 13 2009, from http://www.garlik.com/.

Geonames (2008). *Geonames*. Retrieved December 19 2008, from http://www.geonames.org/about.html.

Glaser, H., T. Lewy, I. Millard and B. Dowling (2007). *On Coreference and the Semantic Web*. Technical Report, Electronics & Computer Science, University of Southampton. Retrieved 1 June 2009, from http://eprints.ecs.soton.ac.uk/15245/.

Glaser, H., I. Millard and A. Jaffri (2008). *RKBExplorer.com:A Knowledge Driven Infrastructure for Linked Data Providers*. In European Semantic Web Conference, LNCS, 5021, 797-801.

Goodman, D (1987). The Complete Hypercard Handbook. Bantam Books, New York.

Gridinoc, L., M. Sabou, M. D'Aquin, M. Dzbor and E. Motta (2008). *Semantic Browsing with PowerMagpie*. European Semantic Web Conference, 802-806.

Grishman, R. (1997). "Information extraction: Techniques and challenges". In Maria Pazienza editions, Information Extraction, Springer-Verlag.

Grønbæk, K., N. O. Bouvinand and L. Sloth (1997). *Designing Dexter-based hypermedia services for the World Wide Web*. ACM Conference on Hypertext, 146-156.

Gruber, T. R (1993). "A translation approach to portable ontologies". Knowledge Acquisition, 5(2), 199-220.

Halasz, F. G., T. P. Moran and R. H. Trigg (1987). *Notecards in a nutshell*. In Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface, 45-52.

Hall, W., H. Davis and G. Hutchings (1996). Rethinking Hypermedia: The Microcosm Approach. Kluwer Academic Publishers.

Handschuh, S., S. Staab, and A. Maedche (2001). *CREAM - Creating relational metadata with a component-based, ontology-driven annotation framework*. In Proceeding of the International Conference on Knowledge Capture.

Handschuh, S. S. Staab, and F. Ciravegna (2002). *S-CREAM – Semi-automatic CREAtion of Metadata*. In European Conference on Knowledge Acquisition and Management.

Hastrup, T., R. Cyganiak and U. Bojars (2008). *Browsing Linked Data with Fenfire*. Linked Data on the Web Workshop, WWW2008.

Heflin, J., J. Hendler and S. Luke (1999). *SHOE: A Knowledge Representation Language for Internet Applications*. Technical Report CS-TR-4078 (UMIACS TR-99-71), Department of Computer Science, University of Maryland.

Henze, N. and W. Nejdl (2001). "Adaptation in Open Corpus Hypermedia". IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems, 12.

Henze, N. And W. Nejdl (2002). *Knowledge Modeling for Open Adaptive Hypermedia*. In International Conference on AH and Adaptive Web-Based Systems.

Hothi, J., and W. Hall (1998). *An Evaluation of Adapted Hypermedia Techniques using Static User Modelling*. In Proceedings of the Second Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, USA, 45-50.

Huynh, D., S. Mazzocchi, and D. Karger (2005). *Piggy Bank: Experience the Semantic Web Inside Your Web Browser*. In International Semantic Web Conference, LNCS, 3729, 413-430.

Hyvonen, E., M. Junnila, S. Kettula, E. Makela, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen (2004). *Publishing Museum Collections on the Semantic Web – The MuseumFinland Portal*. In Proceedings of World Wide Web Conference.

Hyvonen, E., M. Junnila, S. Kettula, E. Makela, S. Saarela, M. Salminen, A. Syreeni, A. Valo, and K. Viljanen (2004b). *MuseumFinland – Finnish Museums on the Semantic Web User's Perspective*. In Proceedings of Museums and the Web.

Hyvonen, E., M. Salminen, M. Junnila and S. Kettula (2004c). *A Content Creation Process for the Semantic Web*. In Proceeding of Ontologies and Lexical Resources in Distributed Environments.

IEEE PAPI (2008). *IEEE Public and Private Information (PAPI) for learners*. Retrieved December 19 2008, from edutool.com/papi.

iGoogle (2008). Retrieved December 19 2008, http://www.google.com/ig.

IMS LIP (2008). *IMS Learner Information Package Specification (LIP)*. Retrieved December 19 2008, from www.imsglobal.org.

Jeffries, R., Miller, J.R., Wharton, C., and Uyeda, K.M. (1991). *User Interface Evaluation in the Real World: A Comparison of Four Techniques*. In Proceedings of the ACM Computer/Human Interaction Conference, 119-124.

Jena (2008). *Jena – Semantic Web Framework*. Retrieved December 19 2008, from jena.sourceforge.net/.

Jena Inference (2008). *Jena 2 Inference Support*. Retrieved December 19 2008, from jena.sourceforge.net/inference/index.html.

Jin, Y., S. Decker, and G. Wiederhold (2001). *OntoWebber: Model-Driven Ontology-Based Web Site Management*. The First International Semantic Web Working Symposium (SWWS'01), Stanford University, Stanford, CA.

Kahan, J., M. R. Koivunen, E. Prud'Hommeaux and R. Swick (2001). *Annotea: An Open RDF Infrastructure for Shared Web Annotations*. In Proceedings of the Tenth International World Wide Web Conference, Hong Kong.

Kaplan, C., J. Fenwick and J. Chen (1993). "Adaptive Hypertext Navigation Based on User Goals and Context". User Modelling and User-Adapted Interaction, 193-220.

Kay, J., and A. Lum (2003). *Ontology-based User Modelling for the Semantic Web.* International Conference on User Modeling Workshop: Personalisation for the Semantic Web, 11-19.

Kim, W. (2002). "Personalization: Definition, Status, and Challenges Ahead". Journal of Object Technology, 1(1), 29-40.

Kiryakov, A. B. Popov, D. Ognyanoff, D. Manov, A. Kirilov and M. Goranov (2004). "Semantic annotation, indexing, and retrieval". Journal of Web Semantics. Volume 2, 49-79.

Kobilarov, G. and I. Dickinson, I. (2008). *Humboldt: Exploring Linked Data*. HP technical report, 2008. Retrieved December 19, from http://www.hpl.hp.com/techreports/2008/HPL-2008-23.html?mtxs=rss-hpl-tr.

Kobsa, A. (2001): "Generic User Modeling Systems". User Modeling and User-Adapted Interaction, 11(1-2), 49-63.

Koivunen, M. R. (2005). *Annotea and Semantic Web Supported Collaboration*. Invited talk at Workshop on User Aspects of the Semantic Web, at the Europen Semantic Web Conference.

Kravcik, M., and D. Gasevic (2006). *Adaptive Hypermedia for the Semantic Web. Conference on Hypertext and Hypermedia*. In Proceedings of the Joint International Workshop on Adaptivity, Personalization & the Semantic Web.

Lara, R., S. K. Han, H. Lausen, M. Stollberg, Y. Ding, and D. Fensel (2004). *An Evaluation of Semantic Web Portals*. IADIS Applied Computing International Conference.

Lawless, S. and V. Wade (2006). *Dynamic Content Discovery, Harvesting and Delivery, from Open Corpus Sources, for Adaptive Systems*. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 4018, 445-451.

Lee, J. H., M. H. Kim, Y. J. Lee (1993). "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies". Journal of Documentation, Volume 49, 188-207.

Lei, Y., E. Motta, and J. Domingue (2004). *Modelling Data-Intensive Web Sites With OntoWeaver*. In Proceedings of the International Workshop on Web-based Information System Modelling (WISM 2004), Riga, Latvia.

Lei, Y., E. Motta, and J. Domingue (2004b). *OntoWeaver-S: Supporting the Design of Knowledge Portals*. Engineering Knowledge in the Age of the SemanticWeb, LNCS 3257, 216-230.

Levenshtein, I. V. (1966). “Binary Codes Capable of Correcting Deletions, Insertions and Reversals”. Cybernetics and Control Theory.

Lin, D. (1998). *An Information-Theoretic Definition of Similarity*. In the Proceedings of the Fifteenth International Conference on Machine Learning.

Lippman, A. (1980). *Movie-maps: An application of the optical videodisc to computer graphics*. In Proceedings of the 7th annual conference on Computer graphics and interactive techniques, 32–42.

Maedche, A., S. Staab, N. Stojanovic, R. Studer, and Y. Sure (2001). *SEmantic portAL – The SEAL approach*. In Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (eds.), *Spinning the Semantic Web*, 317-359. MIT Press, Cambridge, MA, 2001

Maedche, A., S. Staab, R. Studer, Y. Sure, and R. Volz (2002). *SEAL – Typing Up Information Integration and Web Site Management by Ontologies*. IEEE-CS Data Engineering Bulletin, Special Issue on Organizing and Discovering the Semantic Web.

Maedche A. and V. Zacharias (2002). *Clustering Ontology-based Metadata in the Semantic Web.* In Proceedings of the Joint Conferences 13th European Conference on Machine Learning and 6th European Conference on Principles and Practice of Knowledge Discovery in Databases, Springer, LNAI, Finland, Helsinki, 2002.

Maedche, A. and S. Staab (2002). *Measuring Similarity between Ontologies*. In Proceedings of the International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, LNCS, 2473 251-263.

Makela, E., E. Hyvonen, S. Saarela and K. Viljanen (2004). *ONTOVIEWS – A tool for creating semantic Web Portals*. International Semantic Web Conference, LNCS, 3298, 797-811.

Manber, U., A. Patel, and J. Robinson, J. (2000). “Experience with personalization on Yahoo!” Communications of the ACM. 43(8), 35-39.

Meyrowitz, N. (1986). *Intermedia: The Architecture and Construction of an Object-Oriented Hypemedia System and Applications Framework*. In Proceedings of Conference on Object Oriented Programming Systems Languages and Applications, 186–201.

Mika, P. (2005). “Flink: Semantic Web Technology for the Extraction and Analysis of Social Networks”. Journal of Web Semantics, 3(2).

Molich, R., and J. Nielsen (1990). “Improving a Human-Computer Dialogue”. Communications of the ACM, 33(3), 338-348.

Nelson, T. H. (1965). *Complex information processing: a file structure for the complex, the changing and the indeterminate*. ACM Annual Conference/Annual Meeting, 20th National Conference, 84-100.

Nelson, T. H. (1974). Computer Lib; Dream Machines.

Nielsen, J., and R. Molich (1990). *Heuristic Evaluation of User Interfaces*. In Proceedings of the ACM Computer/Human Interaction Conference, 249-256.

Nielsen, J. (1992). *Finding usability Problems Thorough Heuristic Evaluation*. In Proceedings of the ACM Computer/Human Interaction Conference.

Nielsen, J. (1993). Usability Engineering. Academic Press.

Nielsen, J. (1994): *Enhancing the Explanatory Power of Usability Heuristics*. In Proceedings of the ACM Computer/Human Interaction Conference, 152-158.

Nielsen, J. (1995). Multimedia and Hypermedia: the Internet and Beyond. Academic Press.

Nilsson, M., M. Palmér, J. Brase (2003). *The LOM RDF binding - principles and implementation*. 3rd Annual Ariadne Conference.

ObjectViewer. *ObjectViewer Semantic Web browser.* Retrieved December 19 2008, from http://objectviewer.semwebcentral.org/.

Ounnas, A., I. Liccardi, H. C. Davis, D. Millard and S. A. White (2006). *Towards a Semantic Modeling of Learners for Social Networks*. In International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL) at AH2006.

Pearl, A (1989). *Sun's Link Service: a protocol for open linking*. In the Proceedings of the ACM Conference on Hypertext and Hypermedia, 137-146.

Pietriga, E. (2006). *IsaViz: A Visual Authoring Tool for RDF*. Retrieved December 19 2008, from http://www.w3.org/2001/11/IsaViz/

Pietriga, E., C. Bizer, D. Karger and R. Lee (2006). *Frensel: A Browser-Independent Presentation Vocabulary for RDF*. International Semantic Web Conference, LNCS, 4273, 158-171.

Preece, J. (1993). A Guide to Usability, Human Factors in Computing. Addison Wesley, Chapter 6, 107-121.

Protégé (2008). *Protégé Ontology Editor and Knowledge-base Framework*. Retrieved December 19 2008, from protege.stanford.edu.

Protégé2Jena (2008). Retrieved December 19 2008, from http://semweb.krasu.ru/protege2jena/

Quan, D. and D. R. Karger (2004). *How to Make a Semantic Web Browser*. International World Wide Web Conference, New York.

Rada, R., H. Mili, E. Bicknell, and M. Blettner (1989). ”Development and application of a metric on semantic nets”. IEEE Transaction on Systems,Man, and Cybernetics, 19(1), 17–30.

Razmerita, L., A. Angehrn and A. Maedche (2003). *Ontology-based user modeling for Knowledge Management Systems*. In Proceedings of User Modeling, 213-217.

Reeve, L. and H. Han (2005). *Survey of semantic annotation platforms*. Proceedings of the 2005 ACM symposium on Applied computing.

Resnik, P. (1999). “Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language”. Journal of Artificial Intelligence Research, Volume 11, 95-130.

Reynolds, D, P. Shabajee and S. Cayzer (2004). *Semantic Information Portals*. In International Conference on World Wide Web, 290-291.

Richter, M. M (1992). “Classification and Learning of Similarity Measures”. Studies in Classification, Data and Analysis and Knowledge Organization.

Rossi, G., D. Schwabe and R. Guimaraes (2001). *Designing Personalized Web Applications*. In International World Wide Web Conference, Hong Kong, 275-284.

Rutledge, L., J van Ossenbruggen and L. Hardman (2005). *Making RDF presentable: integrated global and local semantic Web browsing*. In International World Wide Web conference.

Şah, M., and Y. D. Konstantin (2005). “Forecasting Enrollment Model Based on First-Order Fuzzy Time Series”. In the Journal of World Academy of Science, Engineering and Technology, Volume 1, 375-378.

Şah, M., and W. Hall (2007). *Building and Managing Personalized Semantic Portals*. In Proceedings of International World Wide Web Conference (WWW 2007), Banff, Alberta, Canada, 1227-1228.

Şah, M., W. Hall, N. M. Gibbins and D. C. De Roure (2007). *SEMPort – A Personalized Semantic Portal.* In Proceedings of the Eighteenth ACM Conference on Hypertext and Hypermedia, 31-32, Manchester, United Kingdom.

Şah, M., W. Hall and D. C. De Roure (2008). *Designing a Personalized Semantic Web Browser*. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS, 5149, 333-336, Springer-Verlag Berlin Heidelberg, Hannover, Germany.

Şah, M., W. Hall and D. C. De Roure (2008b). *SemWeB: A Semantic Web Browser for Supporting the Browsing of Users using Semantic and Adaptive Links*. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems Doctoral Consortium, LNCS, 5149, 431-436, Springer-Verlag Berlin Heidelberg, Hannover, Germany.

Şah, M., W. Hall and D. C. De Roure (2009). *SemWeB Semantic Web Browser – Improving Browsing Experience with Semantic and Personalized Information and Hyperlinks*. 1st National Symposium for Humanities and Technology, Southampton, UK.

Scerri, S. C. Abela and M. Montebello (2005). *semantExplorer: A Semantic Web Browser*. IADIS International Conference WWW/Internet.

Schiaffino S. and A. Amandi (2004). "User – interface agent interaction: personalization issues". International Journal of Human-Computer Studies. 60(1), 129-148.

Schraefel, mc, N. R. Shadbolt, N. Gibbins, S. Harris and H. Glaser (2004). *CS AKTive space: representing computer science in the semantic web*. In International World Wide Web Conference.

Schraefel, mc., D. A. Smith, A. Owens, A. Russell, C. Harris and M. L. Wilson (2005). *The evolving mSpace platform: leveraging the Semantic Web on the Trail of the Memex*. In Hypertext Conference.

Shadbolt N. R., W. Hall and T. Berners-Lee T. (2006). "The Semantic Web Revisited". IEEE Intelligent Systems, 21(3), 96-101.

Shardanand, U. and P. Maes (1995). *Social information filtering: Algorithms for automating "word of mouth."* In Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems, ACM, New York, 210-217.

Shneiderman, B (1987). *User Interfaces Design for the Hyperties Electronic Encyclopaedia.* In Proceedings of the ACM Hypertext Workshop, 189-194,

Sidoroff, T. and E. Hyvönen (2005). *Semantic E-government Portals – A Case Study*. In Proceedings of the ISWC Workshop Semantic Web Case Studies and Best Practices for eBusiness SWCASE05.

SIOC (2008). *SIOC Core Ontology Specification*. Retrieved December 19 2008, from http://rdfs.org/sioc/spec/.

SKOS (2008). *SKOS Core Guide*. Retrieved December 19 2008, from http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.

Speretta, M. and S. Gauch (2005). *Personalized Search Based on User Search Histories*. In IEEE/WIC/ACM international Conference on Web Intelligence.

Spyns, P., D. Oberle, R. Volz, R., J. Zheng, M. Jarrar, M., Y. Sure, R. Studer and R. Meersman (2002). *OntoWeb – A Semantic Web Community Portal*. Fourth International Conference in Practical Aspects of Knowledge Management. LNCS, 2569. Springer Press, 189-200.

Steer, D. (2003). *RDFAuthor*. Retrieved December 19 2008, from, http://rdfweb.org/people/damian/RDFAuthor/.

Sugiyama, K., K. Hatano and M. Yoshikawa (2004). *Adaptive web search based on user profile constructed without any effort from users*. In Proceedings of the 13th International Conference on World Wide Web, 675-684.

Trigg, R. (1983). *A Network-Based Approach to Text Handling for the Online Scientific Community*. PhD Thesis. Retrieved June 8 2009, from http://www.workpractice.com/trigg/thesis-chap4.html.

Turker, A., I. Görgün and O. Conlan (2006). "The Challenge of Content Creation to Facilitate Personalized E-Learning Experiences". International Journal on E-Learning. 5 (1), 11-17. Chesapeake, VA: AACE.

Uren, V.,P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta and F. Ciravegna (2005). "Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art". Journal of Web Semantics, 4(1).

W3 (2009). *Large Triple Stores*. Retrieved 8 June 2009, from http://esw.w3.org/topic/LargeTripleStores.

W3C (1999). *Hypertext Markup Language*. Retrieved 20 December 2008, from http://www.w3.org/MarkUp/html-spec/html-spec_toc.html.

W3C (1999b). *HTML Specification 4.01*. Retrieved 3 June 2009, from http://www.w3.org/TR/html4/.

W3C (2003). *Document Object Model (DOM) Level 2 HTML Specification*, Retrieved December 19 2008, from http://www.w3.org/TR/DOM-Level-2-HTML/.

W3C (2004). *RDF Primer.* Retrieved December 19 2008, from http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

W3C (2004b). *RDF Vocabulary Description Language 1.0: RDF Schema*. Retrieved December 19 2008, from www.w3.org/TR/rdf-schema.

W3C (2004c). *OWL Web Ontology Language Guide*. Retrieved December 19 2008, from www.w3.org/TR/owl-guide.

W3C (2004d). *XML Schema Part 0: Primer Second Edition*. Retrieved May 26 2009, from http://www.w3.org/TR/xmlschema-0/.

W3C (2006). *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. Retrieved December 19 2008, from http://www.w3.org/TR/2006/REC-xml-20060816.

W3C (2007). *Dereferencing HTTP URIs*. Retrieved December 19 2008, from http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14.

W3C (2007b). *Cool URIs for the Semantic Web*. Retrieved December 19 2008, from http://www.w3.org/TR/2007/WD-cooluris-20071217/.

W3C (2008) *SPARQL Query Language for RDF*. Retrieved December 19 2008, from www.w3.org/TR/rdf-sparql-query/.

W3C (2008b). *SRARQL Protocol for RDF.* Retrieved December 19 2008, from http://www.w3.org/TR/rdf-sparql-protocol/.

W3C (2008c). *RIF Use Cases and Requirements*. Retrieved December 19 2008, from http://www.w3.org/TR/rif-ucr/

W3C (2008d). *RDFa Primer. Bringing the Human and Data Webs*. Retrieved 16 February 2009, from http://www.w3.org/TR/xhtml-rdfa-primer/

Weber, G. and P. Brusilovsky (2001). "ELM-ART: An Adaptive Versatile System for Web-based Instruction". International Journal of Artificial Intelligence in Education, 12, 351-384.

Wikipedia (2008). *Project Xanadu*. Retrieved December 19 2008, from http://en.wikipedia.org/wiki/Project_Xanadu.

Wine Ontology (2003). Retrieved May 26 2009, from http://www.w3.org/TR/owl-guide/wine.rdf.

Yesilada, Y., S. Bechhofer and B. Horan (2008). "Dynamic Linking of Web Resources - Customisation and Personalisation". In Advances in Semantic Media Adaptation and Personalization, Springer Series on Studies in Computational Intelligence, Volume 93, 1-24.

Yudelson, M. and P. Brusilovsky (2005). *NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples*. In Proceedings of 12th International Conference on Artificial Intelligence in Education.

Yudelson, M., T. Gavrilova and P. Brusilovsky (2005). *Towards user Modeling Meta-Ontology*. In Proceedings of the International Conference on User Modeling, 448-452, Edinburgh, Scotland, LNAI 3538: Springer, Berlin.

Zhihong, Z. and Z. Mingtian (2003). *A Distributed Description Logic Approach to Semantic Annotation*. International Conference on Parallel and Distributed Computing, Application and Technologies, 219-221, IEEE.

Appendix A. Extra Figures and Codes

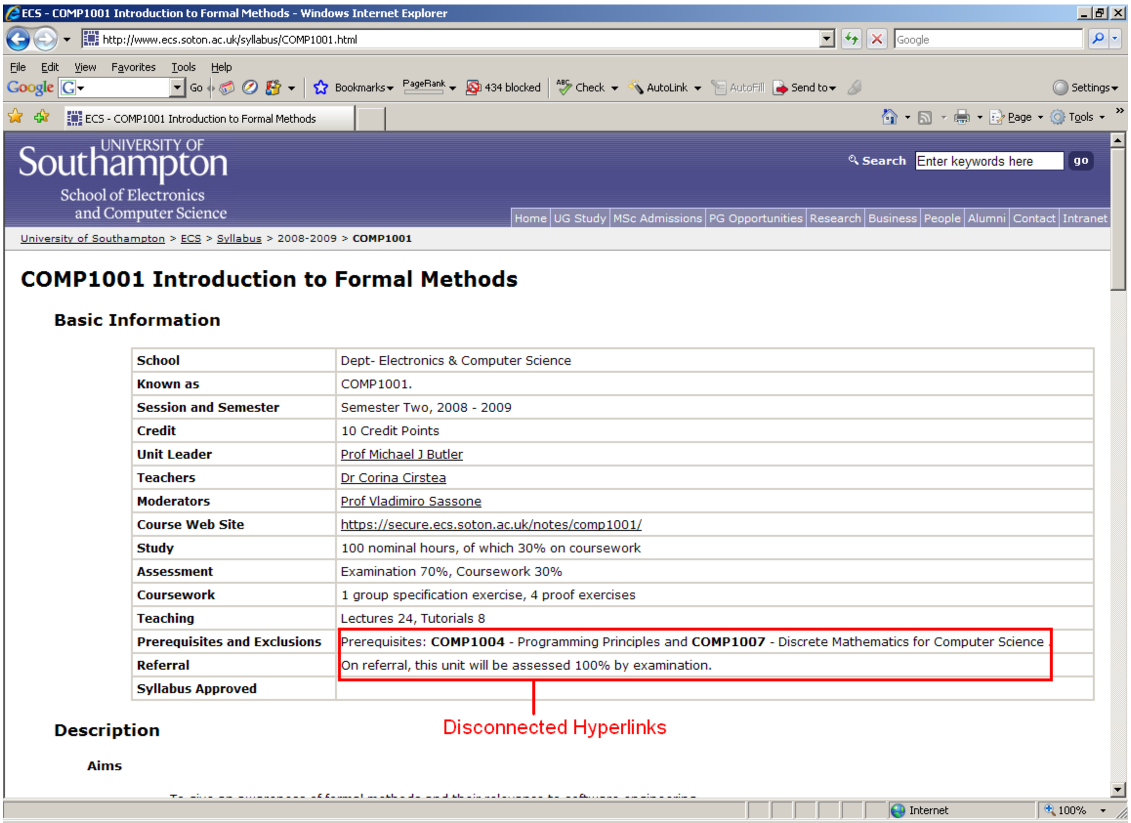


Figure A-1 An example syllabus Web page in the ECS CMWP

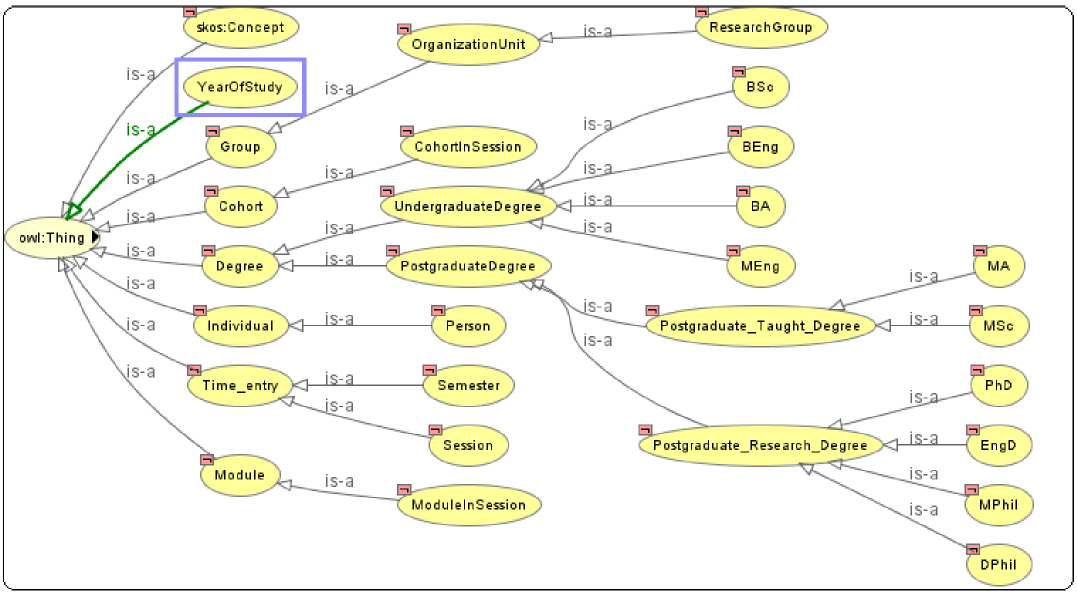


Figure A-2 The visualization of ECS_COURSE classes using OWL Viz

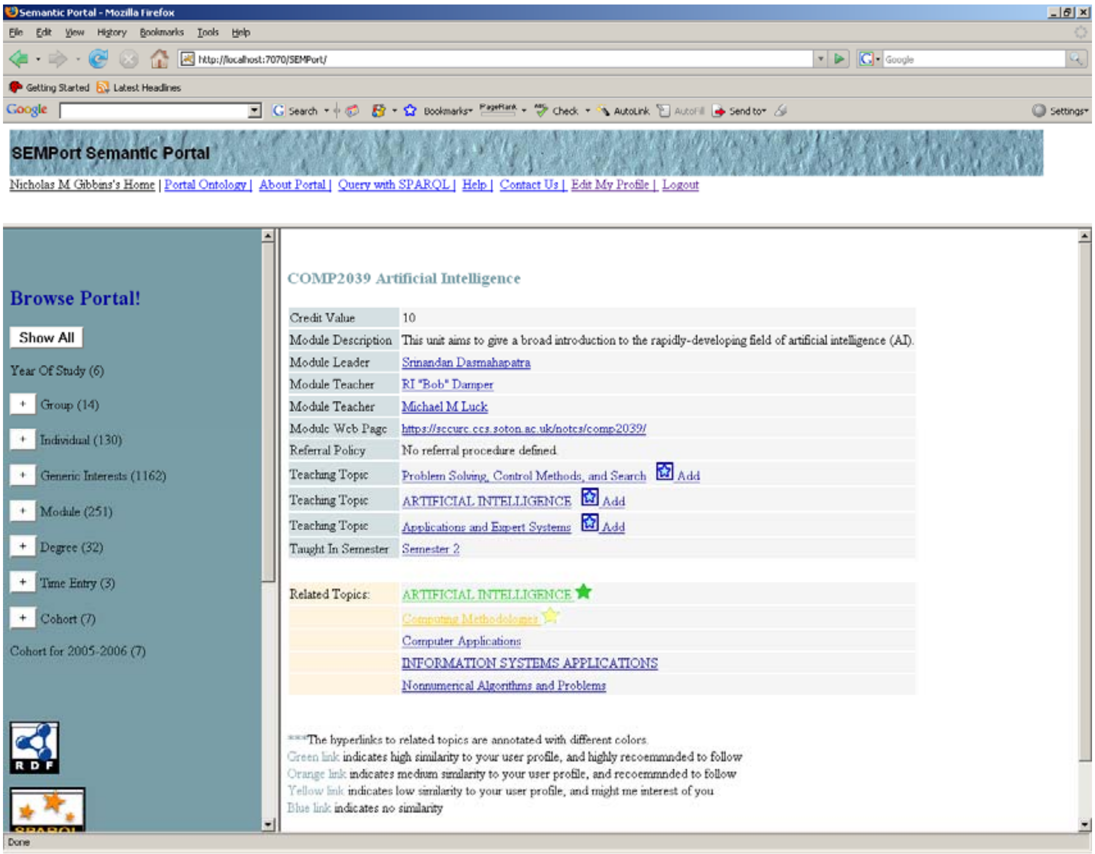


Figure A-3 A view from SEMPort using Mozilla Firefox Web browser

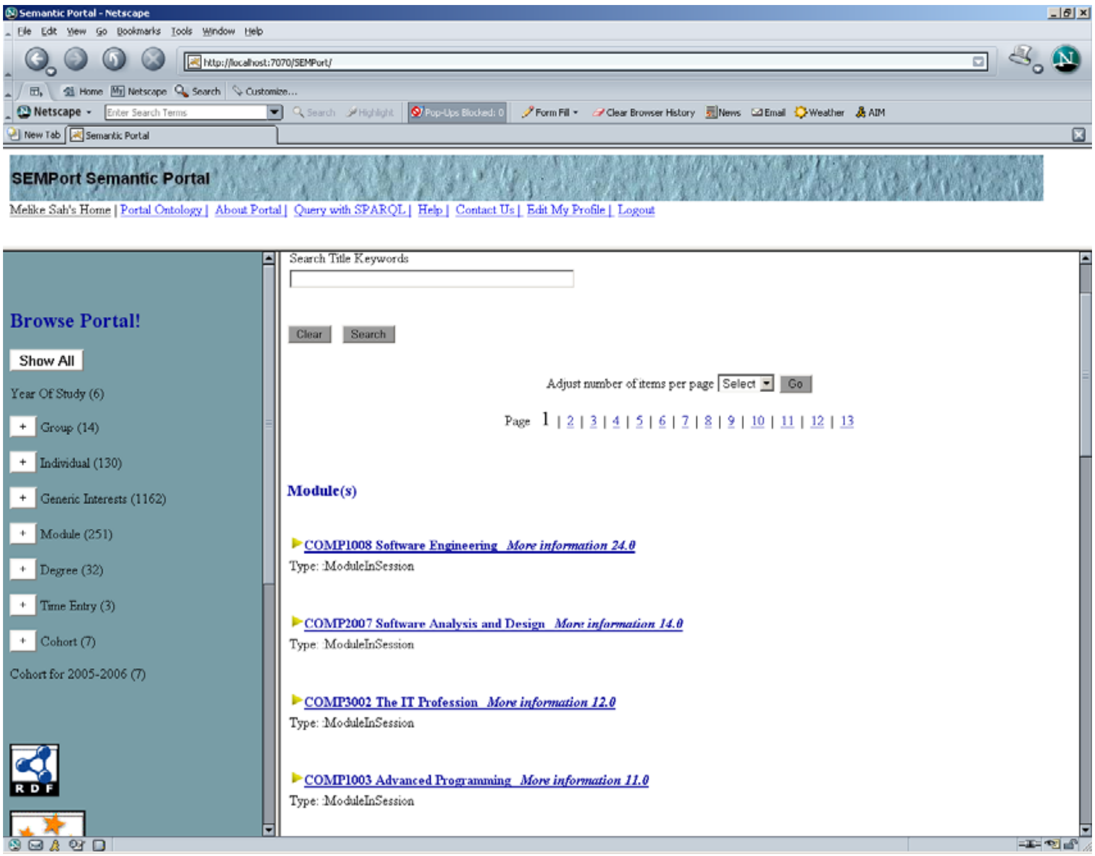


Figure A-4 A view from SEMPort using Netscape Navigator Web browser

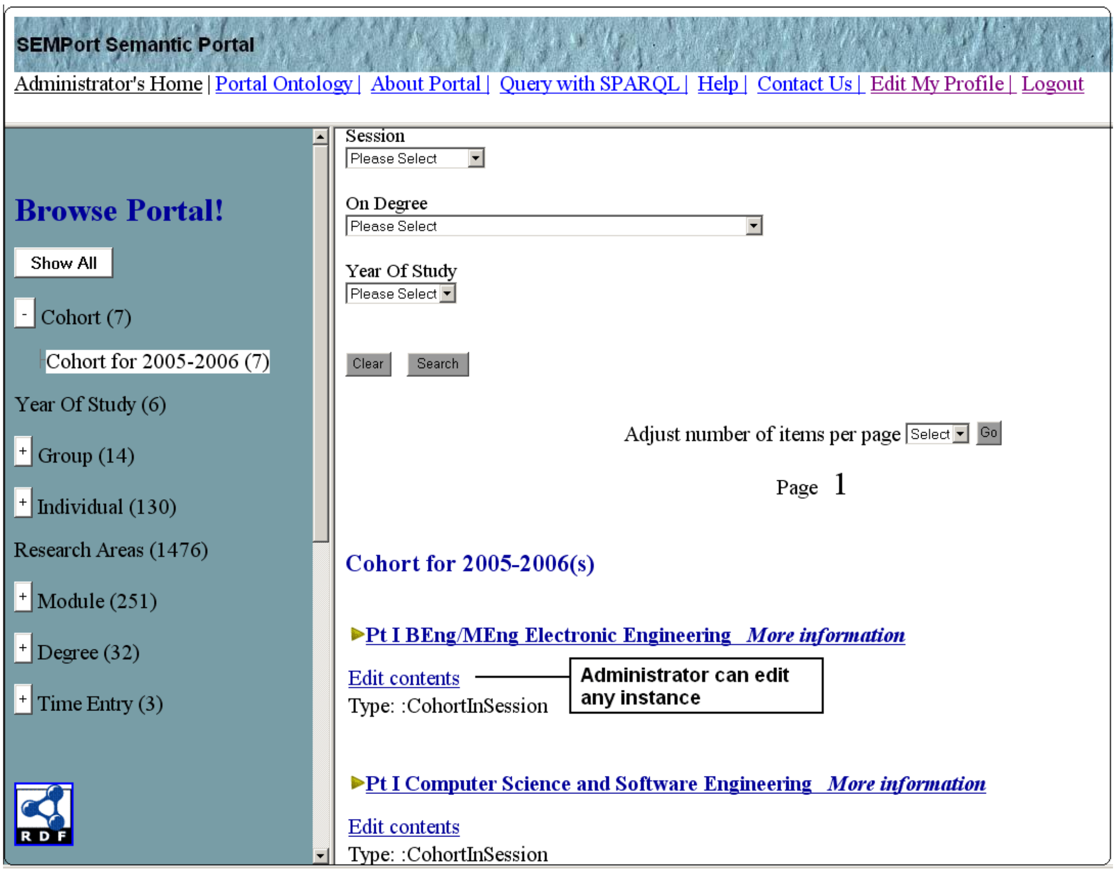


Figure A-5 A view from the semantic navigation. The administrator has access rights to change the contents of the all instances in the portal. In the figure, links are provided with "edit contents"

- **Visibility of system status:** The system should always keep users informed about what is going on.
- **Match between the system and the real world:** The system should speak the user's language rather than system-oriented terms and information presented in natural and logical order.
- **User control and freedom:** When users select a system function by mistake, there should be a clearly marked emergency exit to leave the unwanted state. Support undo/redo.
- **Consistency and standards:** Different words, situations and actions should be presented consistently.
- **Error prevention:** A careful design prevents a problem from occurring
- **Recognition rather than recall:** Make objects, actions and options visible. Instructions for the use of the system should be visible.
- **Flexibility and efficiency of use:** Allow users to tailor frequent actions. Speed up the interaction.
- **Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed.
- **Help users recognize, diagnose, and recover from errors:** Error messages should precisely indicate the problem and constructively suggest a solution.
- **Help and documentation:** It is better to provide help and documentation.

Figure A-6 Nielsen's heuristics with their explanations

Visibility of system status	
Match between system and the real world	
User control and freedom	
Consistency and standards	
Error prevention	
Recognition rather than recall	
Flexibility and efficiency of use	
Aesthetic and minimalist design	
Help users recognize, diagnose, and recover from errors	

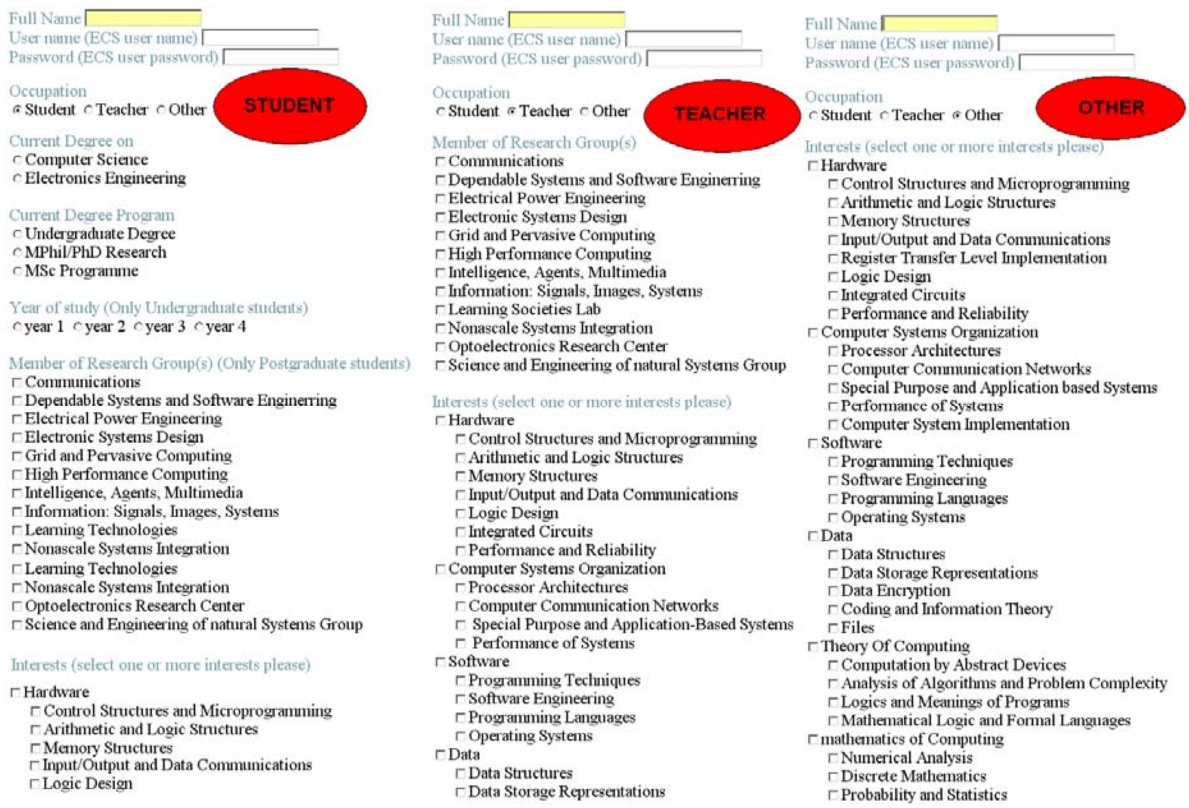
Figure A-7 An example of a structured review form that was used by the reviewers to find out the usability problems

```
String DB_DRIVER = "com.mysql.jdbc.Driver"; //jdbc driver to connect MySQL DB (the KB)
      try{ Class.forName( DB_DRIVER );
      }catch(Exception e) {out.println("Driver expection: " + e);} //check the driver

String DB_URL="jdbc:mysql://localhost:3306/jena_db"; //connection URL and database name
      String DB_USER="melike";  //user name
      String DB_PASSWD="*****"; //password
      String DB_TYPE="MySQL";  //databse type

//create IDB Connection to the DB
IDBConnection conn=new DBConnection(DB_URL, DB_USER, DB_PASSWD, DB_TYPE );

//A Model is created to connect to the KB
Model base=null;
try{
  base=ModelRDB.open(conn,"CMWP");   //open the existing model which is named “CMWP”  from
  the DB }catch (Exception open) {out.println("cannot open");}

//create an ontology model using the base model
OntModel model = ModelFactory.createOntologyModel( OntModelSpec.OWL_DL , base);

//Create a resource for configuring the reasoner
Resource config =  ModelFactory.createDefaultModel().createResource();
//set reasoner to hybrid, which means both forward and backward reasoning rules can be used
//this property can be set to forward or backward as well
config.addProperty(ReasonerVocabulary.PROPruleMode, "hybrid");
//Add the rule’s file to the reasoner
config.addProperty(ReasonerVocabulary.PROPruleSet,  "c://portal_data/portal.rules");

// Create an instance of a generic rule-based reasoner using the configuration explained
above
Reasoner reasoner = GenericRuleReasonerFactory.theInstance().create(config);

//Create an inference model using the configured reasoner and the ontology model
InfModel inf=ModelFactory.createInfModel(reasoner, model);
```

Figure A-8 Creation of ontology and inference models using Jena Ontology API

```
@prefix portal: <http://localhost:7070/portalmini/modules.owl#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix owl: <http://www.w3.org/2002/07/owl#>
@prefix skos: <http://www.w3.org/2004/02/skos/core#>

#Different reasoners can be added to the functionality of Jena rule-based reasoner
#@include  <RDFS>.
#@include  <OWLMini>.
@include  <OWLMicro>.  #in SEMPort, OWLMicro reasoner was used
#@include  <OWL>.

#The following are the all rules that are used in SEMPort
#Finding a prerequisite of a prerequisite
[Prerequisite: (?module portal:otherPrerequisite ?pre2) <-
(?module portal:hasPrerequisite ?pre1),
(?pre1 portal:hasPrerequisite ?pre2),
notEqual(?pre1,?pre2), notEqual(?module,?pre2)]

#Recommending similar topics
[Hierarchy:  (?module rdfs:seeAlso ?topic2) <-
(?module portal:hasTopic ?topic1),
(?topic1 skos:broader ?topic2),  notEqual(?topic1,?topic2) ]

[Relations:  (?module rdfs:seeAlso ?topic2) <-
(?module portal:hasTopic ?topic1),
(?topic1 skos:related ?topic2),  notEqual(?topic1,?topic2) ]

#Module Leader is also Module Teacher. This is used to return correct answers at
ontology-based search when searching for module teachers
[Teacher: (?module portal:hasModuleTeacher ?leader) <-
(?module portal:hasModuleLeader ?leader)]
```

Figure A-9 Rules that are used in SEMPort using rule-based reasoner


Full Name
User name (ECS user name)
Password (ECS user password)
Occupation
Student Teacher Other
STUDENT
Current Degree on
Computer Science
Electronics Engineering
Current Degree Program
Undergraduate Degree
MPhil/PhD Research
MSc Programme
Year of study (Only Undergraduate students)
year 1 year 2 year 3 year 4
Member of Research Group(s) (Only Postgraduate students)
Communications
Dependable Systems and Software Engineering
Electrical Power Engineering
Electronic Systems Design
Grid and Pervasive Computing
High Performance Computing
Intelligence, Agents, Multimedia
Information: Signals, Images, Systems
Learning Technologies
Nonascale Systems Integration
Learning Technologies
Nonascale Systems Integration
Optoelectronics Research Center
Science and Engineering of natural Systems Group
Interests (select one or more interests please)
Hardware
Control Structures and Microprogramming
Arithmetic and Logic Structures
Memory Structures
Input/Output and Data Communications
Logic Design

Full Name
User name (ECS user name)
Password (ECS user password)
Occupation
Student Teacher Other
TEACHER
Member of Research Group(s)
Communications
Dependable Systems and Software Engineering
Electrical Power Engineering
Electronic Systems Design
Grid and Pervasive Computing
High Performance Computing
Intelligence, Agents, Multimedia
Information: Signals, Images, Systems
Learning Societies Lab
Nonascale Systems Integration
Optoelectronics Research Center
Science and Engineering of natural Systems Group
Interests (select one or more interests please)
Hardware
Control Structures and Microprogramming
Arithmetic and Logic Structures
Memory Structures
Input/Output and Data Communications
Logic Design
Integrated Circuits
Performance and Reliability
Computer Systems Organization
Processor Architectures
Computer Communication Networks
Special Purpose and Application-Based Systems
Performance of Systems
Software
Programming Techniques
Software Engineering
Programming Languages
Operating Systems
Data
Data Structures
Data Storage Representations

Full Name
User name (ECS user name)
Password (ECS user password)
Occupation
Student Teacher Other
OTHER
Interests (select one or more interests please)
Hardware
Control Structures and Microprogramming
Arithmetic and Logic Structures
Memory Structures
Input/Output and Data Communications
Register Transfer Level Implementation
Logic Design
Integrated Circuits
Performance and Reliability
Computer Systems Organization
Processor Architectures
Computer Communication Networks
Special Purpose and Application based Systems
Performance of Systems
Computer System Implementation
Software
Programming Techniques
Software Engineering
Programming Languages
Operating Systems
Data
Data Structures
Data Storage Representations
Data Encryption
Coding and Information Theory
Files
Theory Of Computing
Computation by Abstract Devices
Analysis of Algorithms and Problem Complexity
Logics and Meanings of Programs
Mathematical Logic and Formal Languages
mathematics of Computing
Numerical Analysis
Discrete Mathematics
Probability and Statistics

Figure A-10. The questions that are asked to the different user groups during the registration to SEMPort

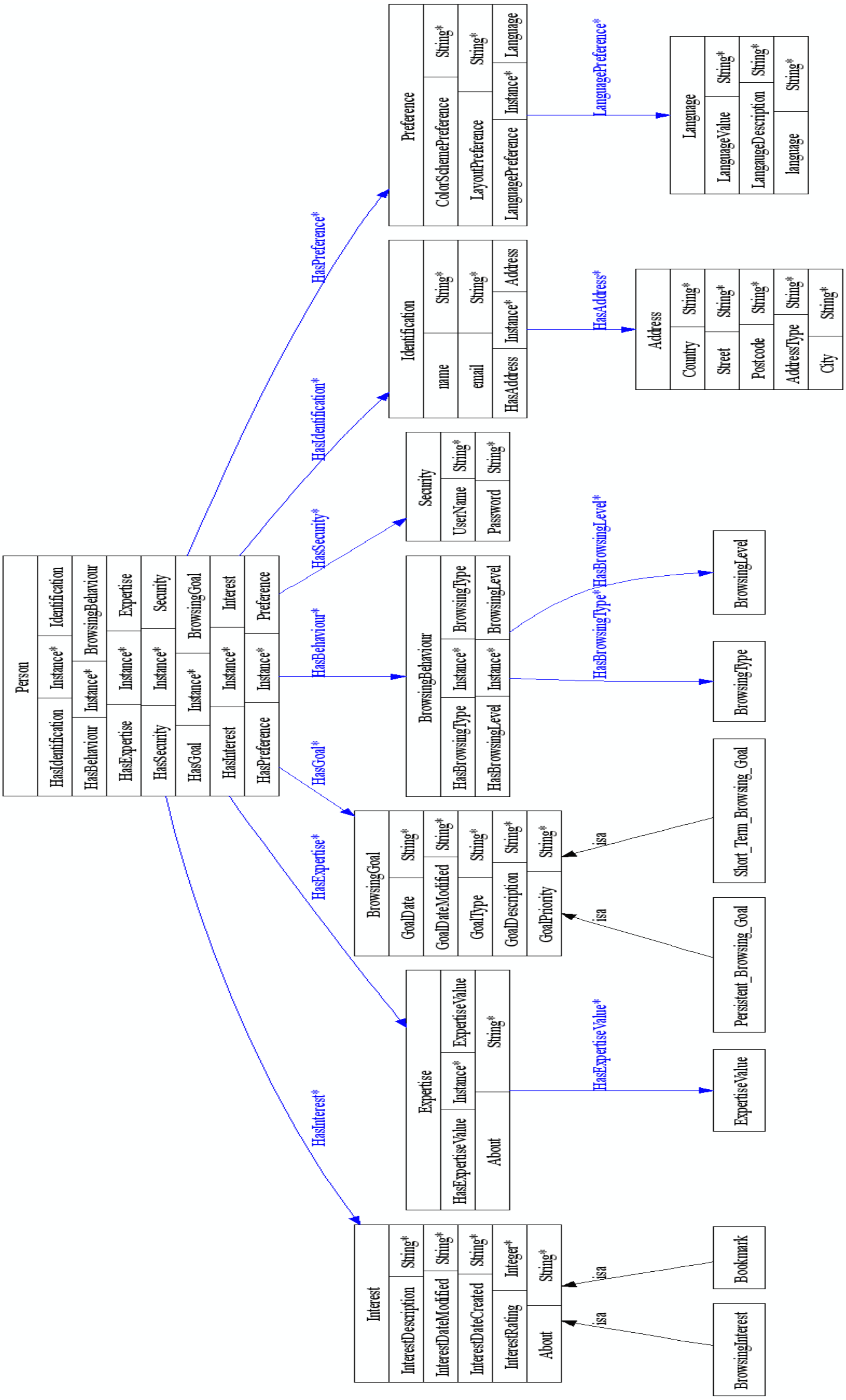


Figure A-11. Concepts used in the proposed user model ontology (represented using Protégé Onto Viz tool)

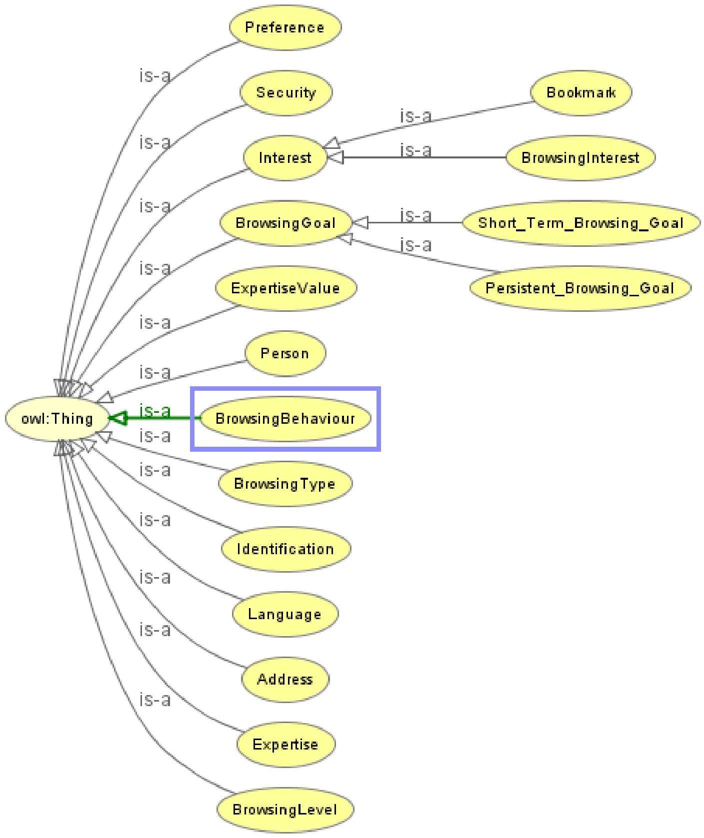


Figure A-12. Is-a diagram showing the hierarchical relationship between the main concepts in the proposed user model ontology

```
OntModel model = ModelFactory.createOntologyModel();
try {
      URL url = new URL(instance_URI);
      URLConnection urlc = url.openConnection();
      //content type that is requested
      urlc.setRequestProperty("Accept", "application/rdf+xml");
      urlc.connect();
      InputStream ins = urlc.getInputStream();
      model.read(ins, instance_URI);
      ins.close();
      }catch (Exception contentnego){testread=false;}
```

Figure A-13. HTTP content negotiation using Jena

```
Rule:DBpediaAnnotation
Priority:200
({Lookup.majorType==DBpedia}):mention
-->
{gate.AnnotationSet mentionSet=(gate.AnnotationSet)bindings.get("mention");
Annotation
mentionAnn=(Annotation)((AnnotationSet)bindings.get("mention")).iterator().next();

String anchor="";
try
{
anchor=(doc.getContent().getContent(mentionAnn.getStartNode().getOffset(),
mentionAnn.getEndNode().getOffset()).toString());
} catch(InvalidOffsetException io){}
com.ontotext.gate.japec.LookupService lookup_mapping=new
com.ontotext.gate.japec.LookupService();

String mapping="";
try
{mapping=lookup_mapping.getMapping("DBpedia_all",anchor);
}catch(Exception mappingexception){System.out.println(mappingexception.toString());}
if (mapping.equals("")|| mapping.equals("empty")){}
else
{FeatureMap features=Factory.newFeatureMap();
features.put("rule", "DBpediaAnnotation");
features.put("class", "DBpedia_class");
features.put("mapping", mapping);
annotations.add(mentionSet.firstNode(), mentionSet.lastNode(),"Mention", features);
annotations.removeAll(mentionSet);
}
}//if no duplicates
```

Figure A-14. Japec rule that is used in DBpedia

Appendix B. SEMPort Questionnaires

Portal Software Evaluation Participation Form and Non-Disclosure Agreement

The following evaluation is in two parts: a hands-on evaluation of some aspects of the software and a follow-up questionnaire. The hand-on evaluation during the tasks will help us understand our interfaces, in order to see errors, misunderstandings and improvements, and the follow-up questionnaire will help us learn if you are satisfied with the interfaces.

Please note we are evaluating the software, not your performance with it. Your feedback will help us improve our work.

Please note also that you can stop at any time during the evaluation. Also, numbers alone are used to identify participant results and so your identity will remain anonymous.

The Evaluation

You will be asked to perform a set of tasks, using the portal software. The evaluator will explain the tasks ahead of time. She will also show you how the software works, and then you will be given time to try it out yourself before the actual test. You will then be asked if you are ready to begin.

Once a task starts, the evaluator will encourage you to think aloud and you will be able to ask questions while performing the tasks.

Taking Part

By participating in this experiment you are doing your own choice.

Non-Disclosure Agreement

Participant	Witness
Participant ID ________________________	Name ____________________________
Signature ____________________________	Signature _________________________
Date ________________________________	Date _____________________________

NIELSEN'S 10 HEURISTICS USED AT THE STRUCTURED REVIEW:

1. **Visibility of system status**
   The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. **Match between system and the real world**
   The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in natural and logical order.

3. **User control and freedom**
   Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. **Consistency and standards**
   Users should not have to wonder whether different words, situations, or actions men the same thing. Follow platform conventions.

5. **Error prevention**
   Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

6. **Recognition rather than recall**
   Make objects, actions and options visible. The user should not have to remember information from on part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. **Flexibility and efficiency of use**
   Accelerators - unseen by the novice user - may often speed up the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design**
   Dialogues should not contain information which irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. **Help users recognize, diagnose, and recover from errors**
   Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. **Help and documentation**
    Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

EVALUATION TASKS

Thank you for your help in conducting our research!

Group A

Task 1

- Assume this year, you will take the course COMP2004 and you are willing to find out module leader and module teachers.
- Find the module leader and module teacher(s) of course COMP2004.

Task 2

- Assume you are 1st year BEng Electronic Engineering student and you want to find out which courses are compulsory for this year's cohort.
- List the course codes of compulsory modules for "I BEng Electronic Engineering" cohort.

Task 3

- Assume you want to take some courses, but you did not take COMP1003 before and you want to make sure that the courses you wanted do not have COMP1003 as a prerequisite.
- Find the courses which have prerequisite COMP1003.

Group B

Task 1

- Assume this year, you will take the course COMP1007 and you are willing to find out module leader and module teachers.
- Find the module leader and module teacher(s) of course COMP1007.

Task 2

- Assume you are 2nd year BEng Electronic Engineering student and you want to find out which courses are compulsory for this year's cohort.
- List the course codes of compulsory modules for "II BEng Electronic Engineering" cohort.

Task 3

- Assume you want to take some courses, but you did not take COMP1004 before and you want to make sure that the courses you wanted do not have COMP1004 as a prerequisite.
- Find the courses which have prerequisite COMP1004.

For All Groups

Task 4

- Assume you are in second semester and you are willing to take courses which teach artificial intelligence.
- Find courses that teach “artificial intelligence” and taught in “semester 2”

Task 5

- Assume you want to take a course, which is named speech processing and you want to check out that you took all the prerequisites of the course (including the prerequisite of prerequisites)
- Find the course, which is name “speech processing”, and list all the prerequisites of this course

Task 6

- Assume in the previous semester you took COMP1008 and you really liked this course. Now, you want to find other courses, which might be relevant to this course.
- Find course “COMP1008” and then list other courses, which might be relevant to this course (i.e. modules that teach similar or same topics)

Task 7

- In this task you are free to do two tasks of your own.
- Do two tasks of your own (for example, find courses of your interest, find topics of your interest, find teachers and courses that they teach, etc.). Write down these tasks to the papers provided and comment on it.

Task 8

- To analyze the personalization
    1. Register to the personalization (provide some background and personal details).
    2. Login to the portal and open your personalized homepage. Follow some hyperlinks depending on your interest and list some interested hyperlinks to the provided paper.
    3. Open your profile and edit some information using the interface (i.e. change weights of interests, change your name, delete some interests, etc.)
    4. Check that you are logged on to the portal. Then, start navigation again (go to modules, generic area of interest, degree, etc.). Browse the information and this time add some bookmarks using the interface.
    5. Go to user profile and change weights of interests or delete unwanted interests.
    6. Start navigation again, and observe the changes at the navigation

Task 9

- To analyze the content editing interface
    1. Go to personalized homepage and open the contents of the module for update
    2. Add a new module teacher using the interface
    3. Edit some contents (i.e. exam percentage, coursework percentage, title, etc.)
    4. Delete some contents
    5. Upload the changes and go to personalized homepage and open the course contents to see the changes

POST QUESTIONNAIRES

Evaluation of Navigation For Group A
Please circle the most appropriate answer.

*1. I am able to **use** navigation*
Very difficult With some difficulty Unsure Easy Very easy

*2. **How difficult** was it to **find** information using the navigation?*
Very difficult With some difficulty Unsure Easy Very easy

*3. **How well** were you able to **complete tasks** using the navigation?*
Very difficult With some difficulty Unsure Easily Very easy

*4. The **speed** of the navigation was*
Very Slow Slow Fine Fast Very fast

*5. How **useful** did you find presented hyperlinks during the navigation?*
Not at all Not too useful Don't know Useful Very useful

*6. Navigation **improved** my browsing facilities*
Strongly Disagree Disagree Undecided Agree Strongly agree

*7. Overall, **how well** were you **satisfied** with the navigation?*
Very dissatisfied Dissatisfied Undecided Satisfied Very satisfied

---

Please list three **negative** aspect(s):
1.
2.
3.
Please list three **positive** aspect(s):
1.
2.
3.

Additional **Comments** and **Suggestions**:

Evaluation of Navigation For Group B
Please circle the most appropriate answer.

*8. I am able to **use** navigation*
Very difficult With some difficulty Unsure Easy Very easy

*9. **How difficult** was it to **find** information using the navigation?*
Very Difficult With some difficulty Unsure Easy Very easy

*10. **How well** were you able to **complete tasks** using the navigation?*
Very difficult With some difficulty Unsure Easily Very easy

*11. The **speed** of the navigation was*
Very Slow Slow Fine Fast Very fast

*12. How **useful** did you find presented hyperlinks during the navigation?*
Not at all Not too useful Don't know Useful Very useful

*13. Navigation **improved** my browsing facilities*
Strongly Disagree Disagree Undecided Agree Strongly agree

*14. Overall, **how well** were you **satisfied** with the navigation?*
Very dissatisfied Dissatisfied Undecided Satisfied Very satisfied

---

Please list three **negative** aspect(s):
1.
2.
3.
Please list three **positive** aspect(s):
1.
2.
3.

Additional **Comments** and **Suggestions**:

Evaluation of Search
Please circle the most appropriate answer.

*15. **How difficult** was it to **find** information using the search?*
Very difficult Difficult Unsure Easy Very easy

*16. The **speed** of the searches were*
Very Slow Slow Fine Fast Very fast

*17. Overall, **how well** were you **satisfied** with the search and do you **want** this kind of search at ECS Course Modules Web Page?*
Definitely No No Don't know Yes Definitely yes
Please list three **negative** aspect(s):
1.
2.
3.
Please list three **positive** aspect(s):
1.
2.
3.

Additional **Comments** and **Suggestions**:

Evaluation of personalization
Please circle the most appropriate answer.

18. The ***registration*** *was*
Too long   Long   Fair   Short   Too short

19. ***How easily*** *were you able to edit your profile?*
Very difficult   With some difficulty   Unsure   Easy   Very easy

20. ***How easily*** *were you able to add bookmarks?*
Very difficult   With some difficulty   Unsure   Easy   Very easy

21. How ***useful*** *did you find information and hyperlinks on personalized homepage?*
Not at all   Not too useful   Don't know   Useful   Very useful

22. How ***useful*** *did you find reordering of contents at the navigation?*
Not at all   Not too useful   Don't know   Useful   Very useful

23. How ***useful*** *did you find the coloured hyperlinks?*
Not at all   Not too useful   Don't know   Useful   Very useful

24. Overall, ***how well*** *were you* ***satisfied*** *with the personalization and do you* ***want*** *this kind of personalization at ECS Course Modules Web Page?*
Definitely No   No   Don't know   Yes   Definitely yes

Please list three **negative** aspect(s):
1.
2.
3.
Please list three **positive** aspect(s):
1.
2.
3.

Additional **Comments** and **Suggestions**:

Evaluation of Content Editing Interface
Please circle the most appropriate answer.

25. How difficult was it to ***change*** *information?*
Very difficult   Difficult   Unsure   Easy   Very easy

26. How difficult was it to ***add*** *new information?*
Very difficult   Difficult   Unsure   Easy   Very easy

27. How difficult was it to ***delete*** *information?*
Very difficult   Difficult   Unsure   Easy   Very easy

28. The ***speed*** *of the update was*

Very Slow Slow Fine Fast Very fast

29. Overall, ***how well*** *were you* ***satisfied*** *with the content editing interface?*
Very dissatisfied Dissatisfied Undecided Satisfied Very satisfied
Please list three **negative** aspect(s):
1.
2.
3.
Please list three **positive** aspect(s):
1.
2.
3.

Additional **Comments** and **Suggestions**:

Overall reaction to System A
Please circle the most appropriate answer.
32. Enjoyed using the system
Strongly disagree Disagree Don't know Agree Strongly agree

33. Grade the system usability
Not useful at all Not useful Fair Useful Very Useful

34. Grade the usability of tests
Not useful at all Not useful Fair Useful Very Useful

35. Would you prefer to use the system in the future?
No Yes

Overall reaction to System B
Please circle the most appropriate answer.
36. Enjoyed using the system
Strongly disagree Disagree Don't know Agree Strongly agree

37. Grade the system usability
Not useful at all Not useful Fair Useful Very Useful

38. Grade the usability of tests
Not useful at all Not useful Fair Useful Very Useful

39. Would you prefer to use the system in the future?
No Yes

40. Which feature of the system did you like most?
Links Search Personalization Content Edition None

Appendix C. SEMPort Walkthroughs

Assume Sue is a computer science undergraduate student at the School of Electronics and Computer Science (ECS) and she browsing ECS modules using SEMPort. In addition, she has logged into SEMPort for personalized contents.

Scenario 1: Sue accessed to her personalized homepage from SEMPort. In this page, hyperlinks to related modules and ACM CCS topics are presented according to her interests and background. ACM CCS topics are also annotated with visual cues according to her interests as shown in Figure C-1. When she clicked onto ELEC2018 module from the personalized homepage and a detailed view of ELEC2018 is presented to her (Figure C-2). At the bottom of the module Web page, more related topics are presented and annotated with visual cues according to the her interests. From this page, she clicked onto “software” topic and detailed view of this topic is presented to Sue as shown in Figure C-3. She wanted to know more about “programming languages” topic and clicked onto it. In the new page, a list of programming languages courses are presented to her using inverse links (Figure C-4), such as “Advanced Programming”, “C programming”, “Scripting Languages”, etc. She found these links interesting and followed to some of them.

Scenario 2: In the second scenario, Sue wants to find modules taught in semester 2, so that she can decide which courses she can select. She used the ontology hierarchy for this purpose. She selected “Time Entry” concept (see Figure C-5) and from here she clicked on “semester 2” and found all taught modules in semester 2 using inverse links as shown in Figure C-6.

Scenario 3: Sue browsing the modules using ontology hierarchy. Since she is logged into SEMPort, modules are ordered according to her interests. As can be seen in Figure C-7, programming languages courses has the highest rank and presented at the top of the page. Sue found these modules useful and followed for their details.

Scenario 4: Sue wants to find modules taught by "Less Carr" and uses ontology-based search to find answers as shown in Figure C-8. Less Carr is module leader of COMP1003 and COMP3016 and a query on module teacher will return no results without using inferencing. With the use of inference rules we provided in the inference module, module leaders are also added as module teachers to the KB in SEMPort. In Figure C-9, it is seen that modules COMP1003 and COMP3016 are presented in the search results. Then, Sue clicked onto COMP3016. In Figure C-10, Less Carr is appeared as both module teacher and module leader of COMP3016.

Scenario 5: Sue added a bookmark to "hypertext and hypermedia" topic during browsing (from Figure C-10). After bookmark addition, the recommended links are highlighted with different visual cues (Figure C-11). Then, she clicked onto modules from the ontology hierarchy. As can be seen in Figure C-12, modules are reordered according to the new interest, such that "multimedia systems" and "hypertext and hypermedia technologies" modules are presented at the top of the page to Sue. Sue liked the adaptability of the portal to her interests. Finally, Sue wanted to see her profile and clicked onto profile editor from the portal. She updated some interest ratings from this interface as illustrated in Figure C-13.

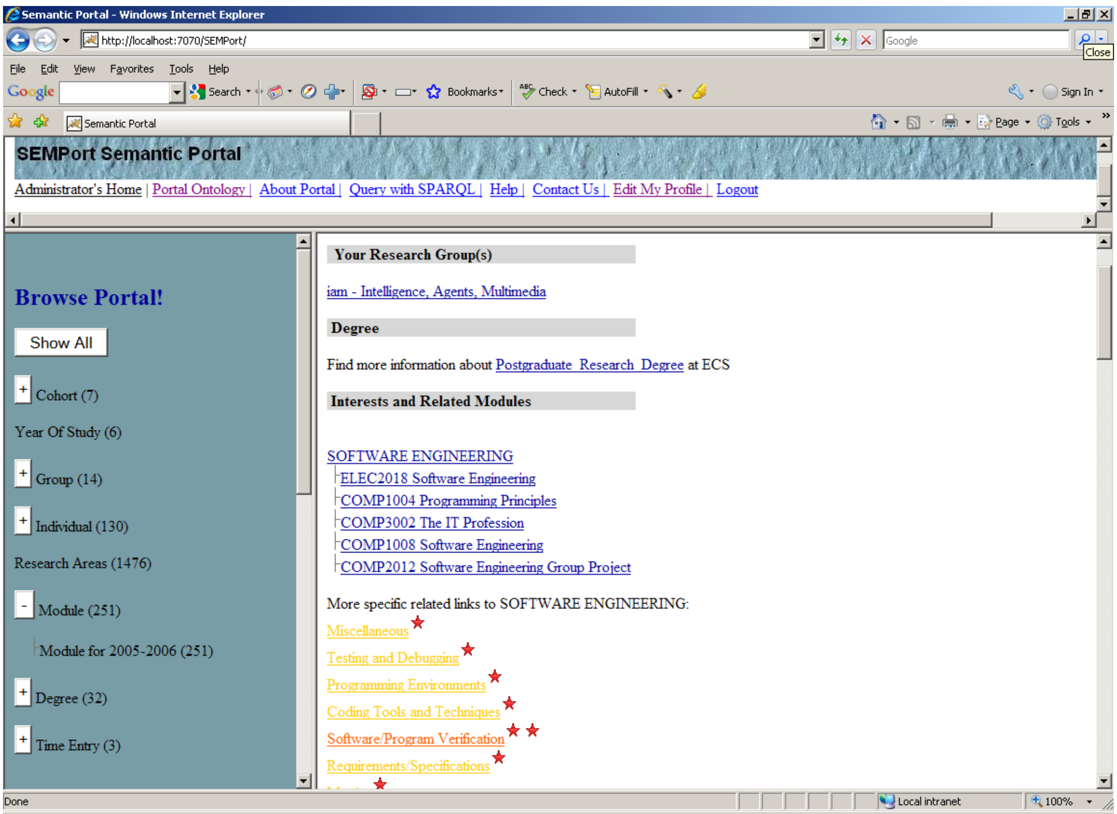


Figure C-1. Personalized homepage of Sue

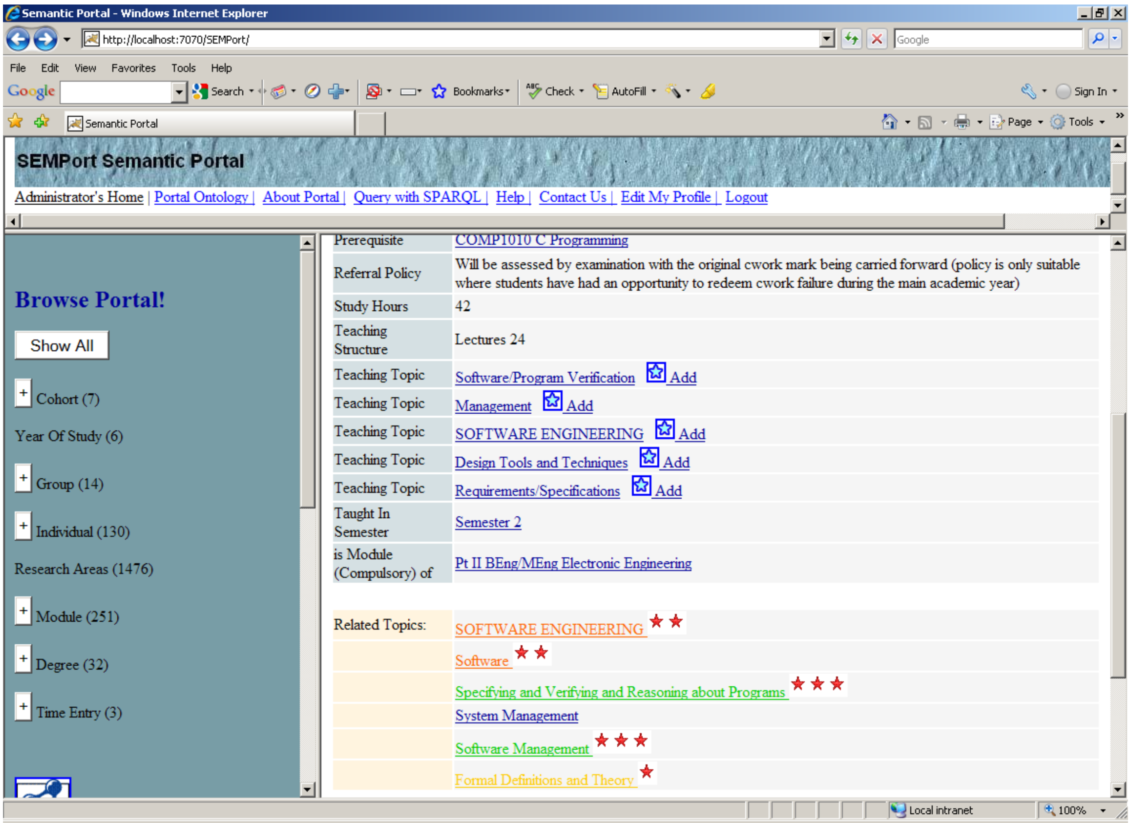


Figure C-2. Detailed view of ELEC2018 with semantic and personalized information and hyperlinks

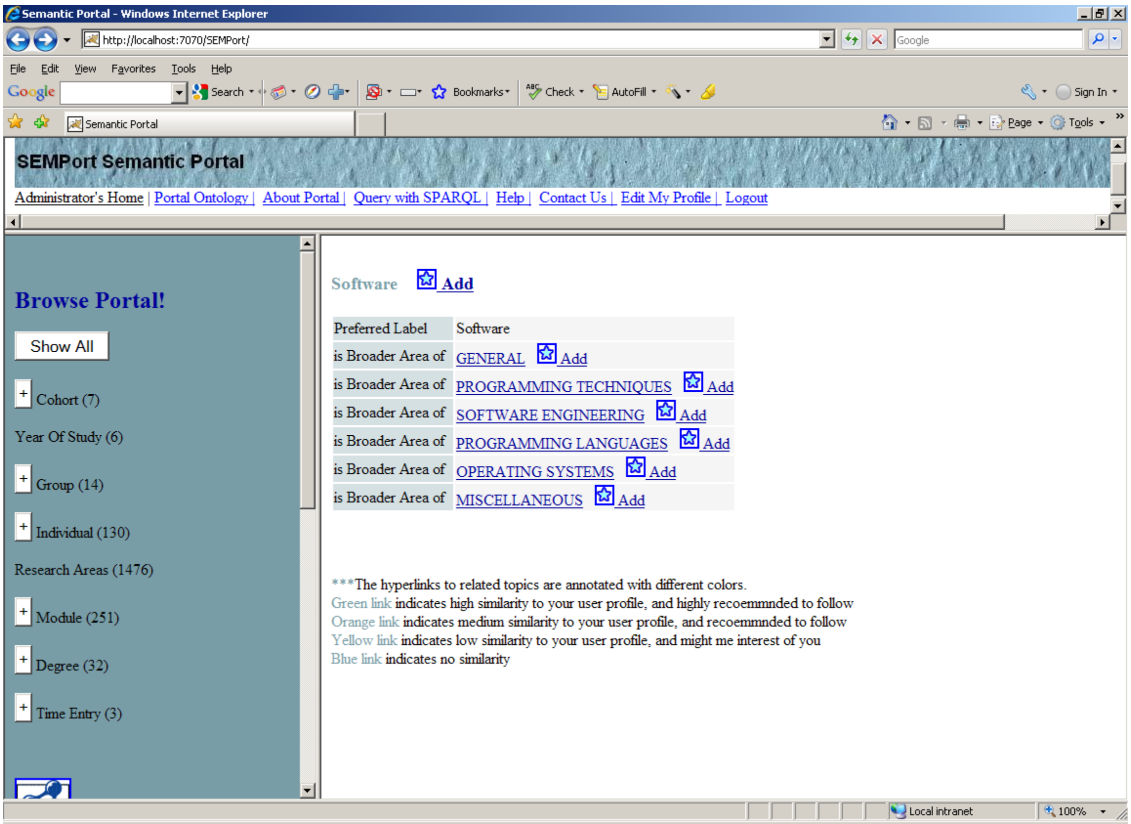


Figure C-3. Detailed view of “Software” ACM CCS topic

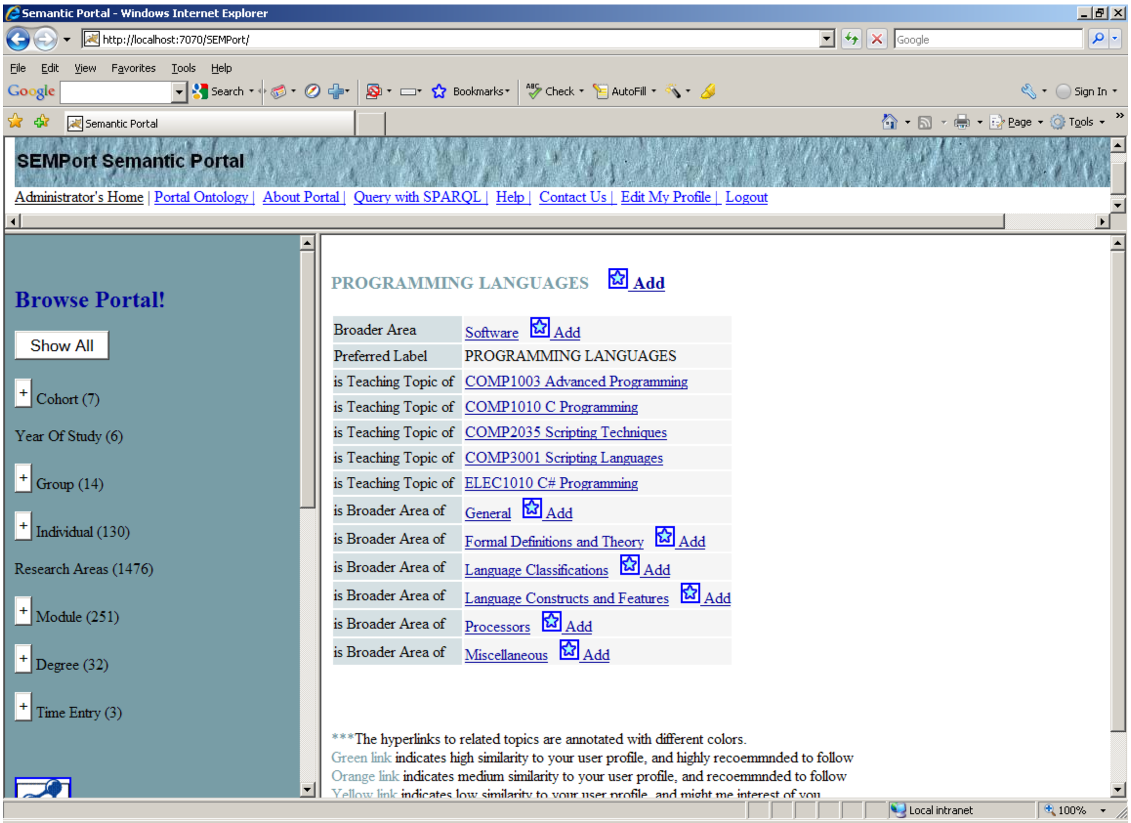


Figure C-4. Detailed view of “Programming Languages” ACM CCS topic

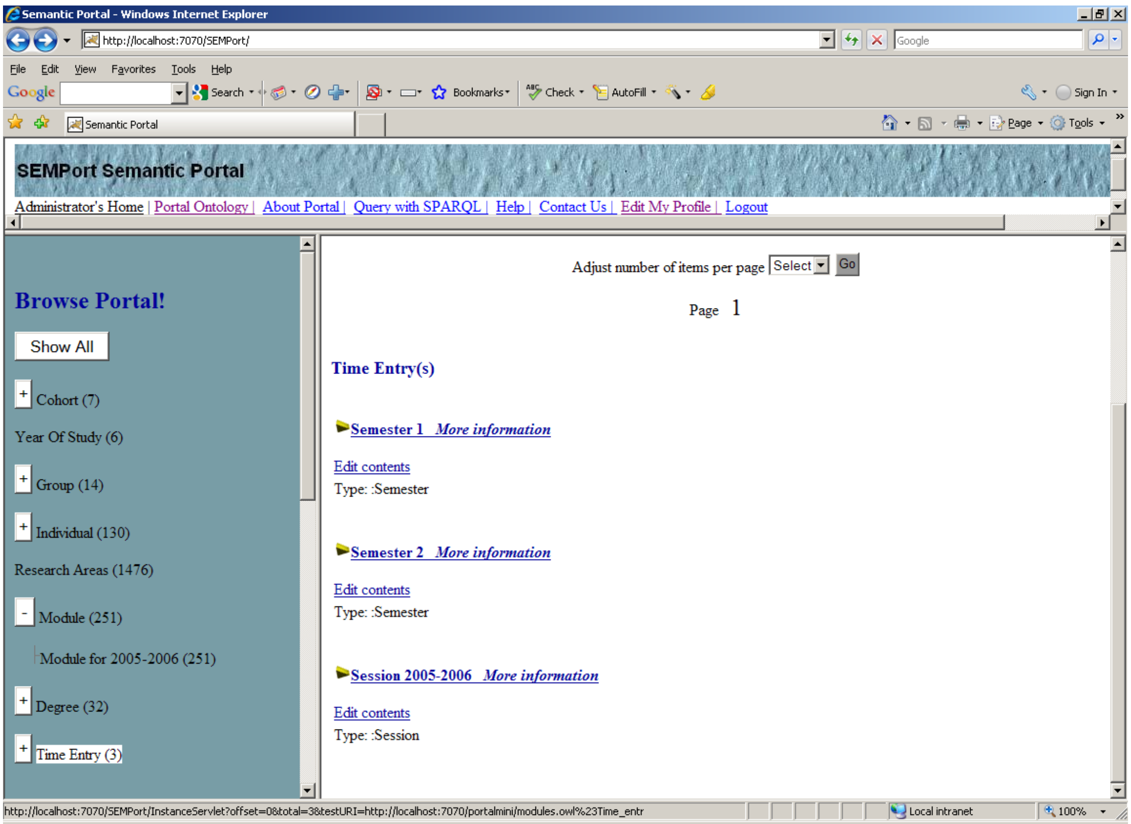


Figure C-5. Sue selected Time Entry concept from the ontology hierarchy

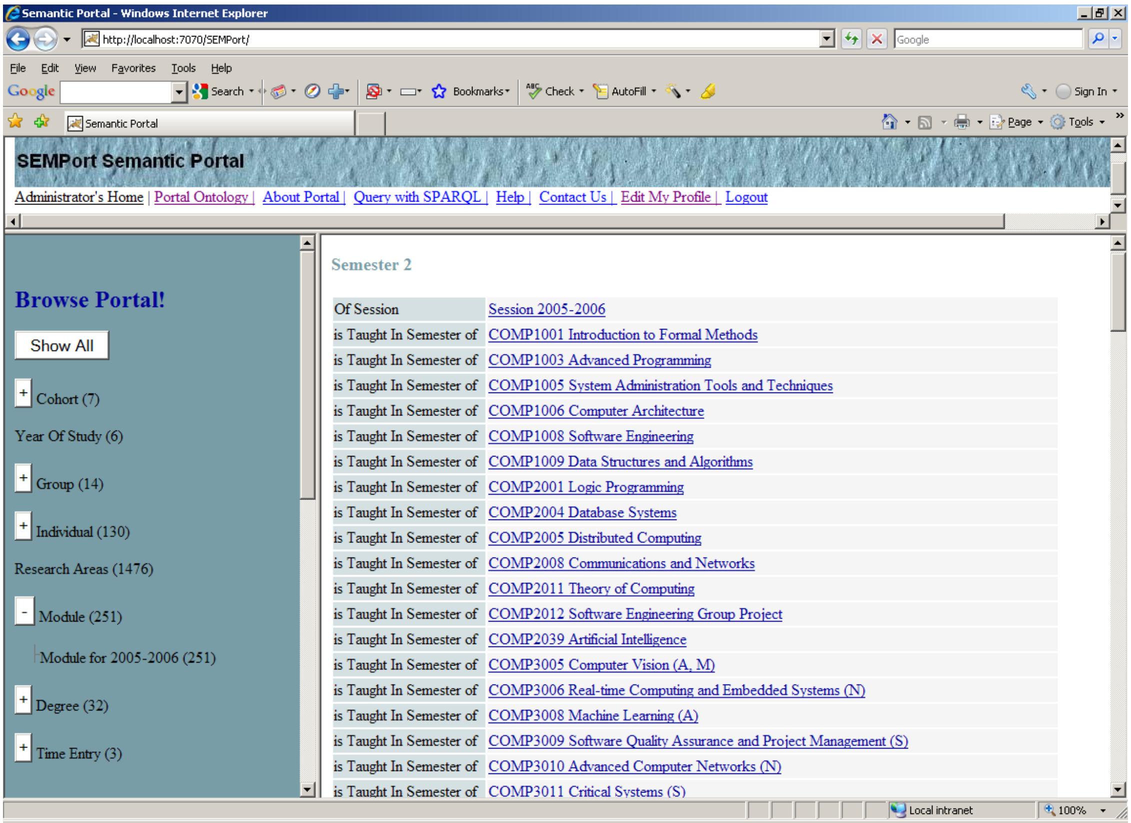


Figure C-6. Detailed view of Semester 2 with all the taught modules in it

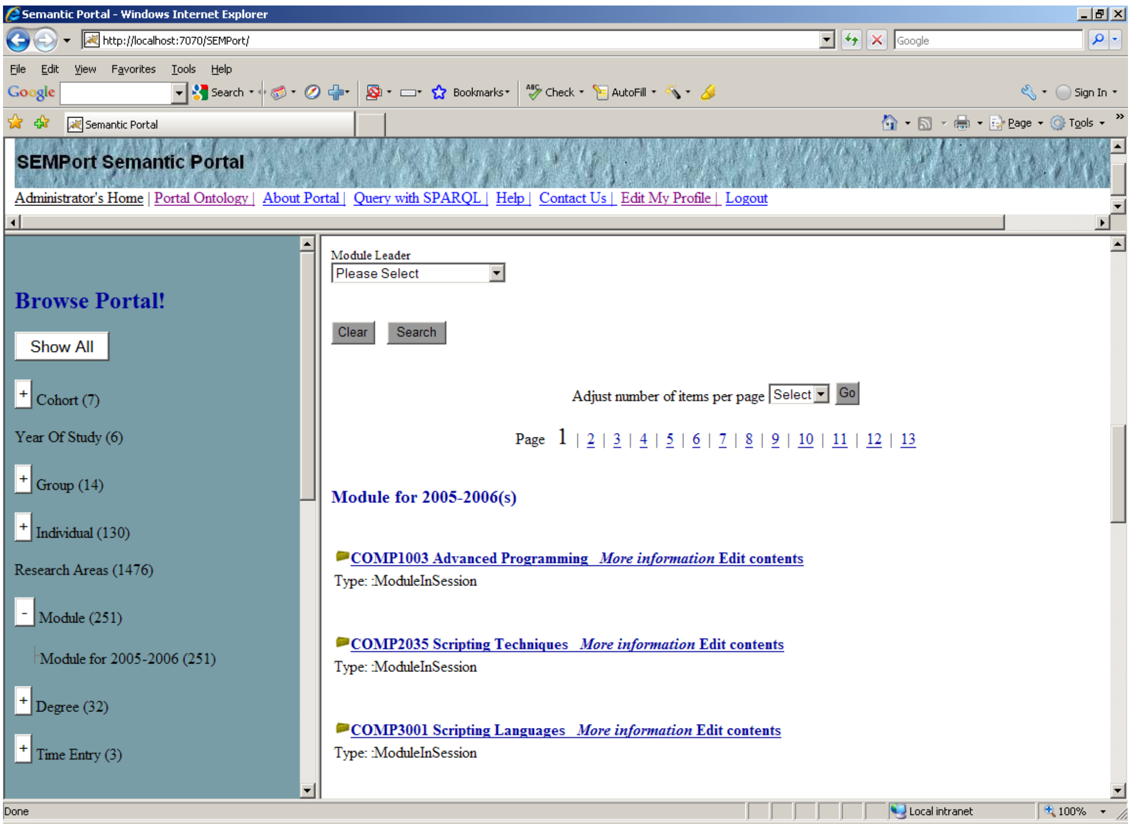


Figure C-7. Modules are ordered according to interests of Sue during the presentation of general view

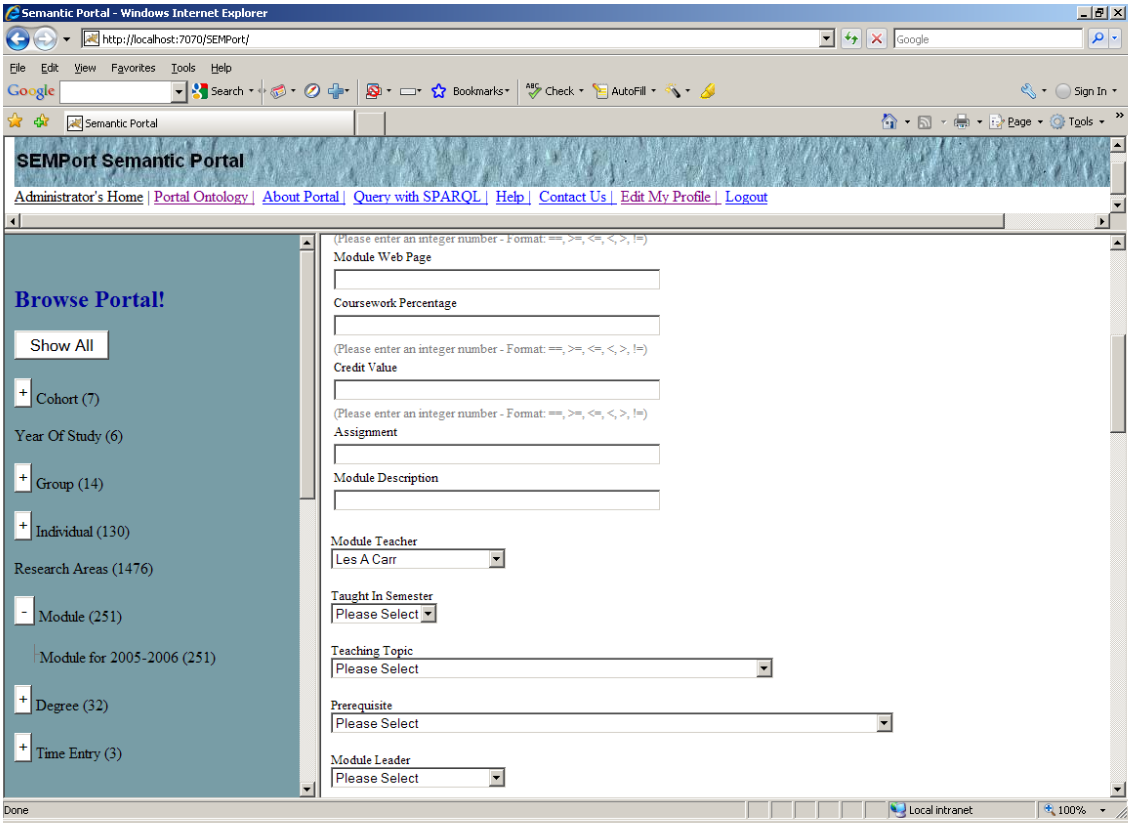


Figure C-8. Sue uses “module teacher” property to find modules taught by “Less Carr”

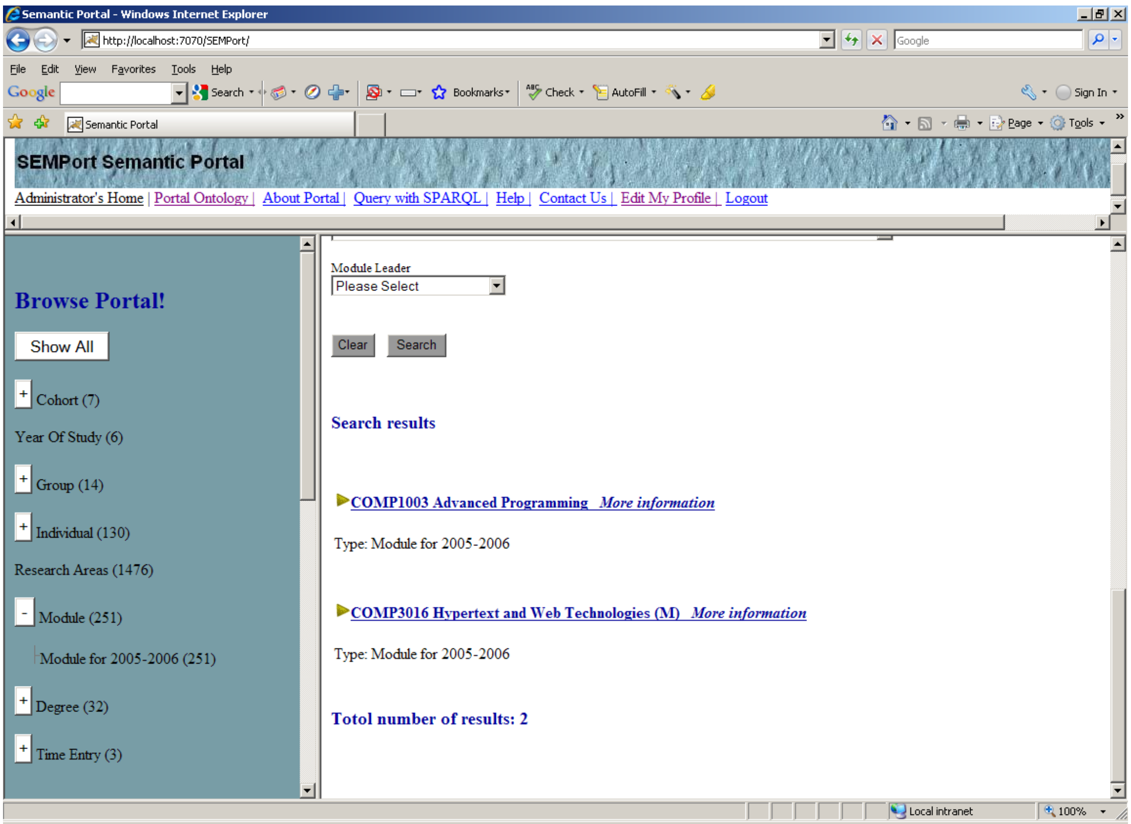


Figure C-9. Search results. Although Less Carr is module leader of both modules, query results display the taught modules by Less Carr accurately using inference

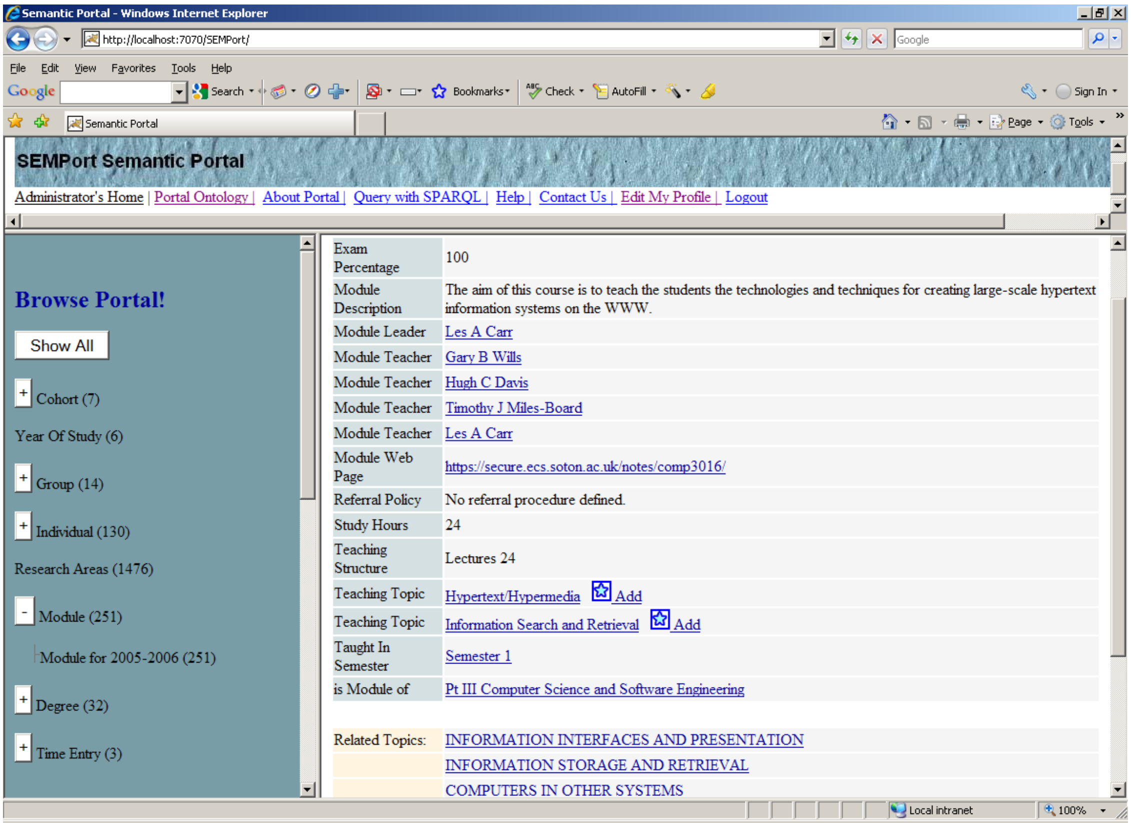


Figure C-10. Detailed view of COMP3016. Less Carr is appeared as leader and teacher of the module

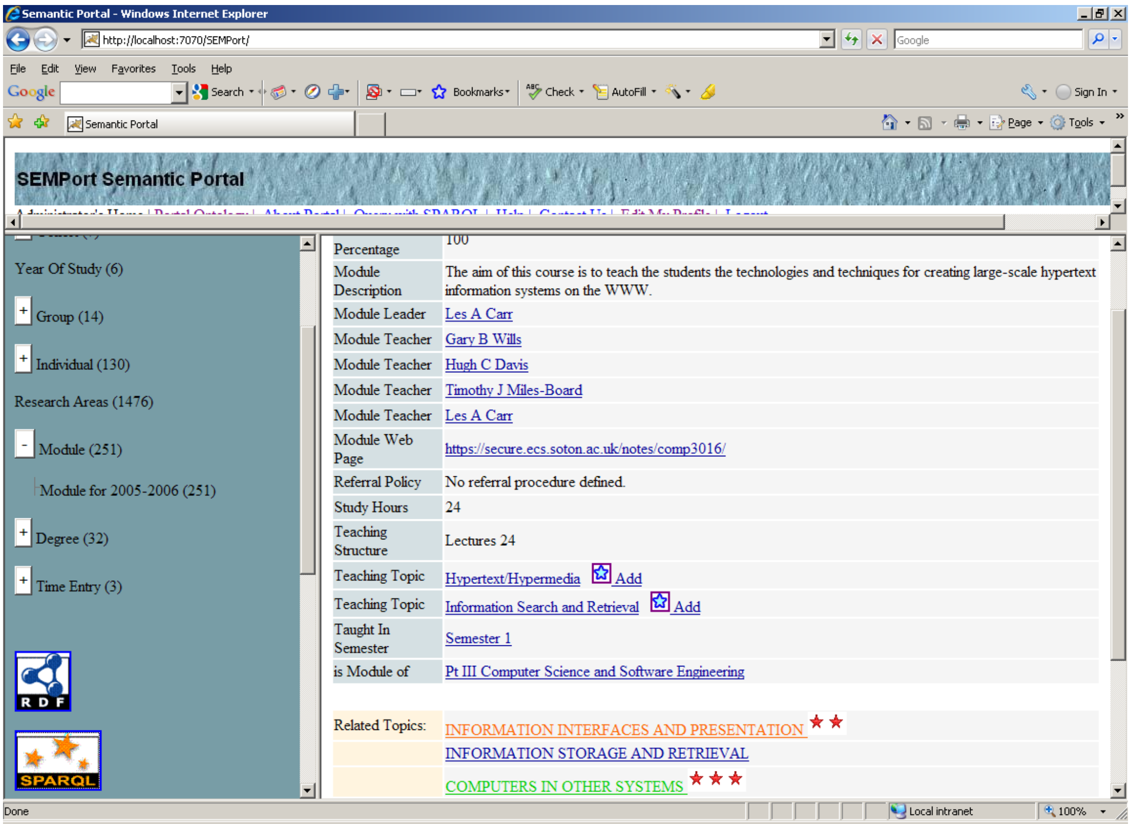


Figure C-11. Detailed view of COMP3016. After bookmark addition, the recommended links are highlighted with visual cues at the bottom of the page.

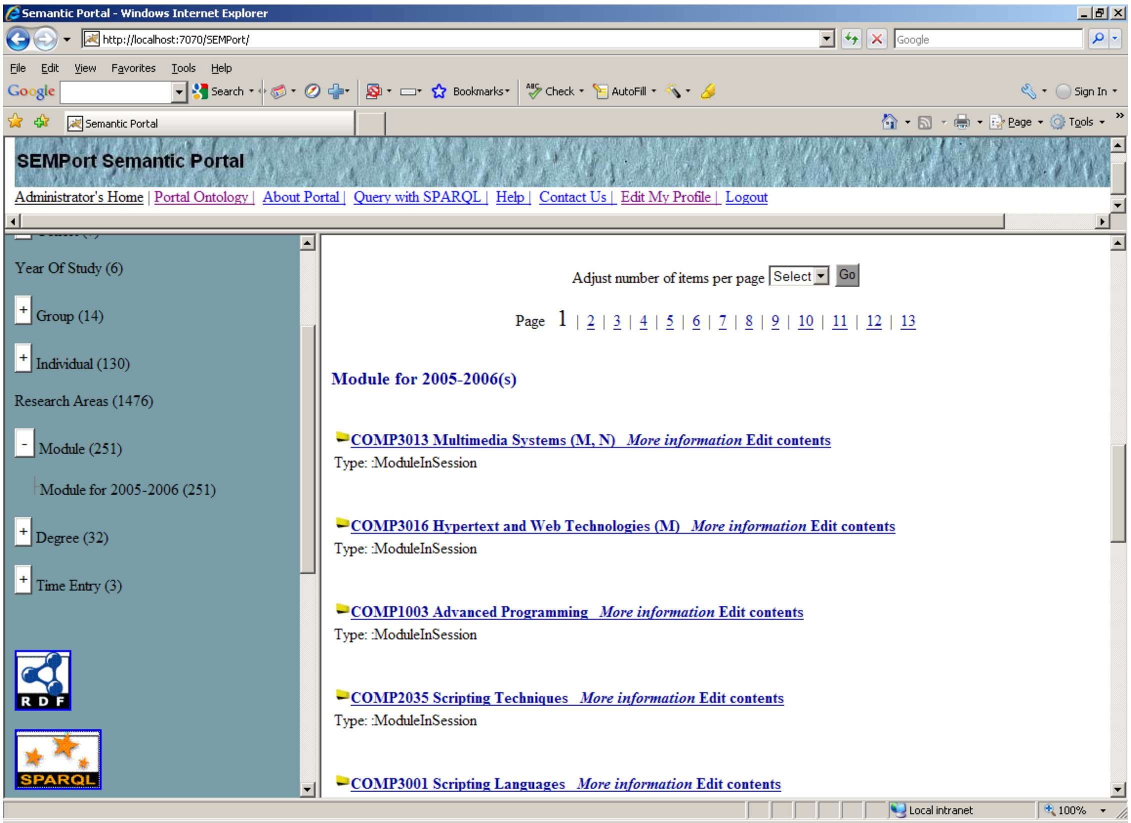


Figure C-12. Modules are ordered based on the new interest (hypertext/hypermedia) of Sue, where “Multimedia Systems” and “Hypertext and Web technologies” became the first two most related modules.

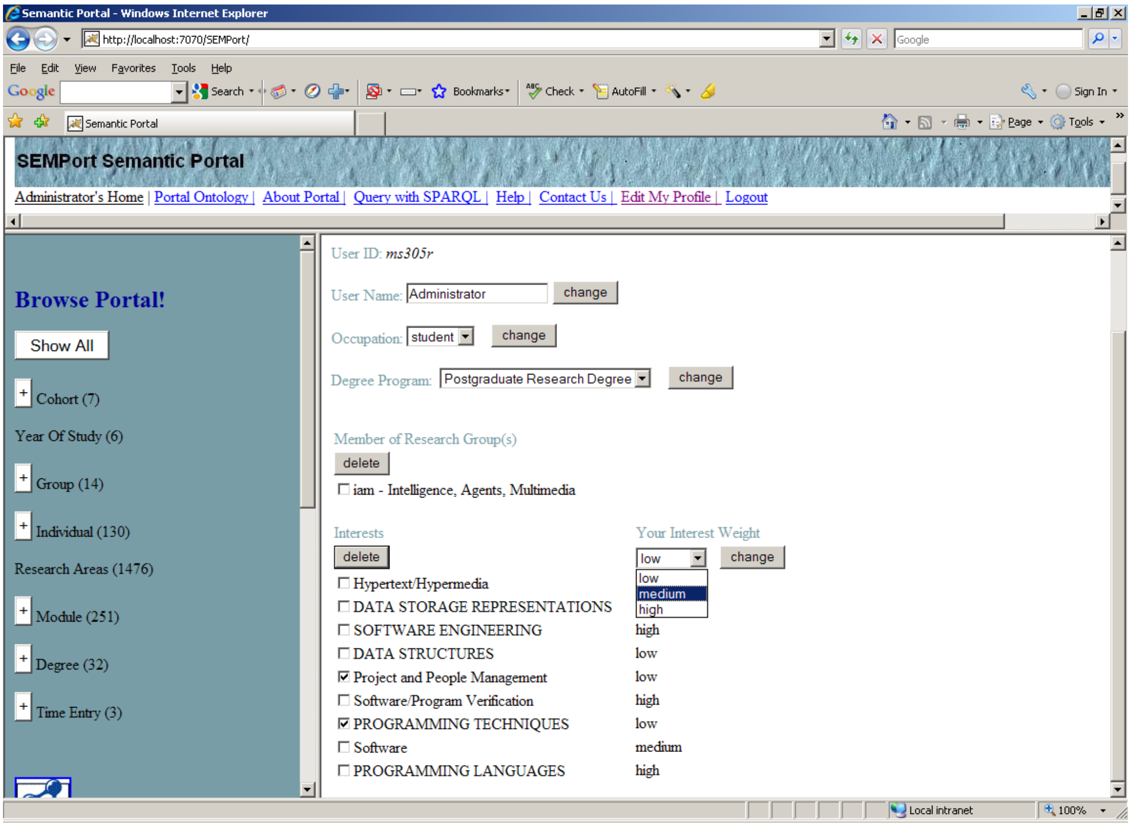


Figure C-13. Profile edition interface, where Sue editing multiple interests at one time.

Appendix D. SEMWeB Walkthroughs

Scenario 1: Assume Sue is a postgraduate student in the School of Electronics and Computer Science (ECS). She is preparing a report about her research subject and related works. For the literature review section of the report, she is researching related publications from the Google. She found a publication, which she thinks it might be related to her research as shown in Figure D-1. She followed through to the ECS eprints website (Figure D-2) and viewed the pdf version of the publication. In addition to this, she wanted to discover more about this page, authors and related topics. For this purpose, she opened the SemWeB sidebar extension, selected the ECS ontology and annotated the Web page as presented in Figure D-3. Then, she requested items of interests using the ontology hierarchy (Figure D-3). For personalized contents, she also logged into SemWeB (Figure D-4 (a) and D-4(b)) and selected some of the provided browsing goals to find more information about interested items as shown in figure D-5. Using the provided links and information, she discovered related people and topics.

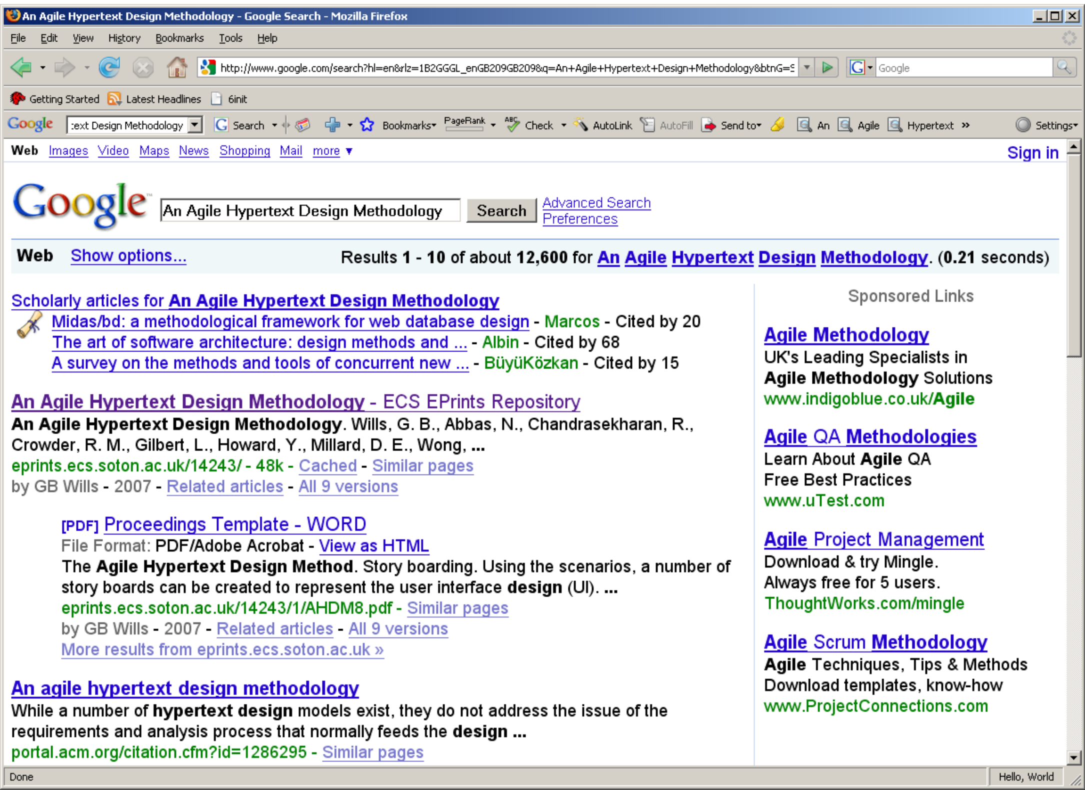


Figure D-1. Sue came across to an interesting publication using Google search

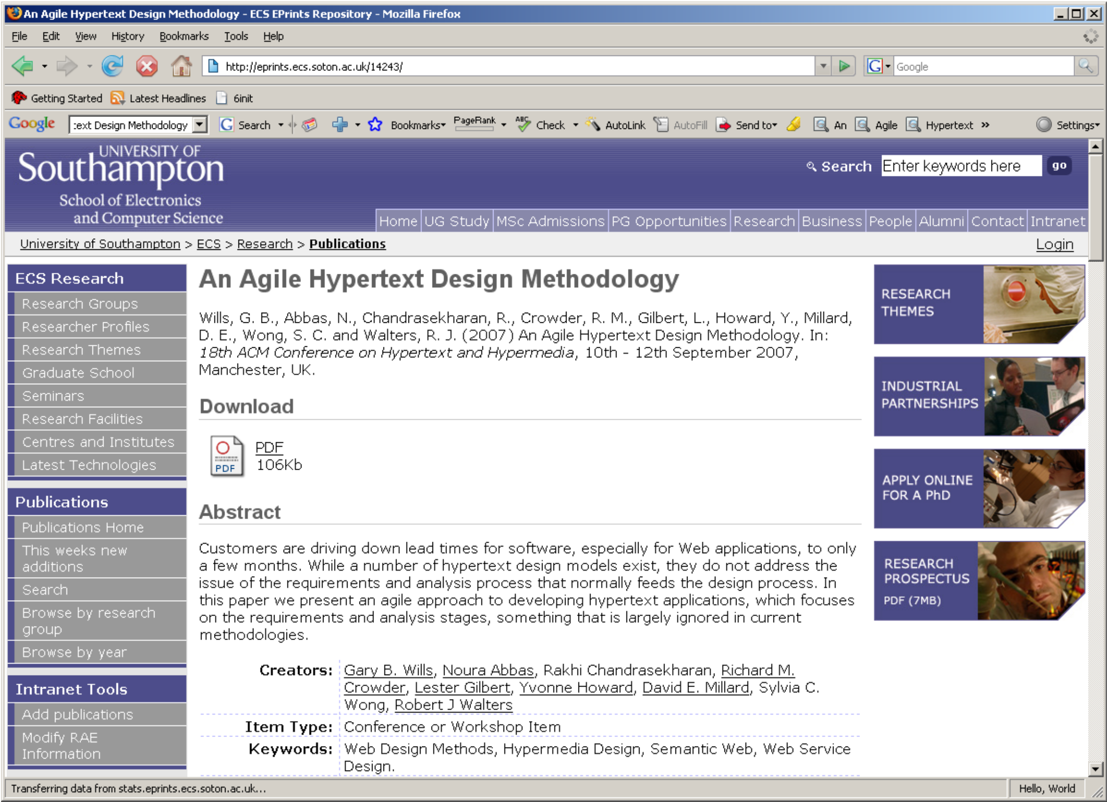


Figure D-2. Sue follows to the Eprints repository from the search results

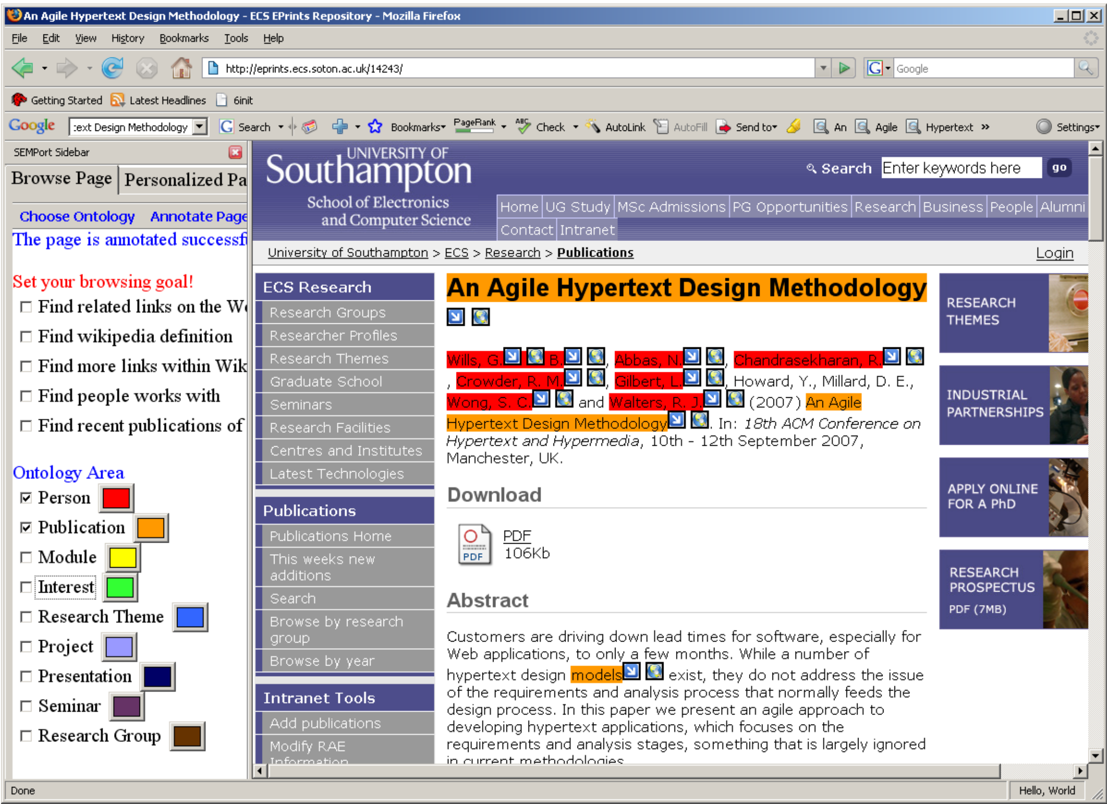


Figure D-3. For more information and links Sue uses the SemWeB sidebar

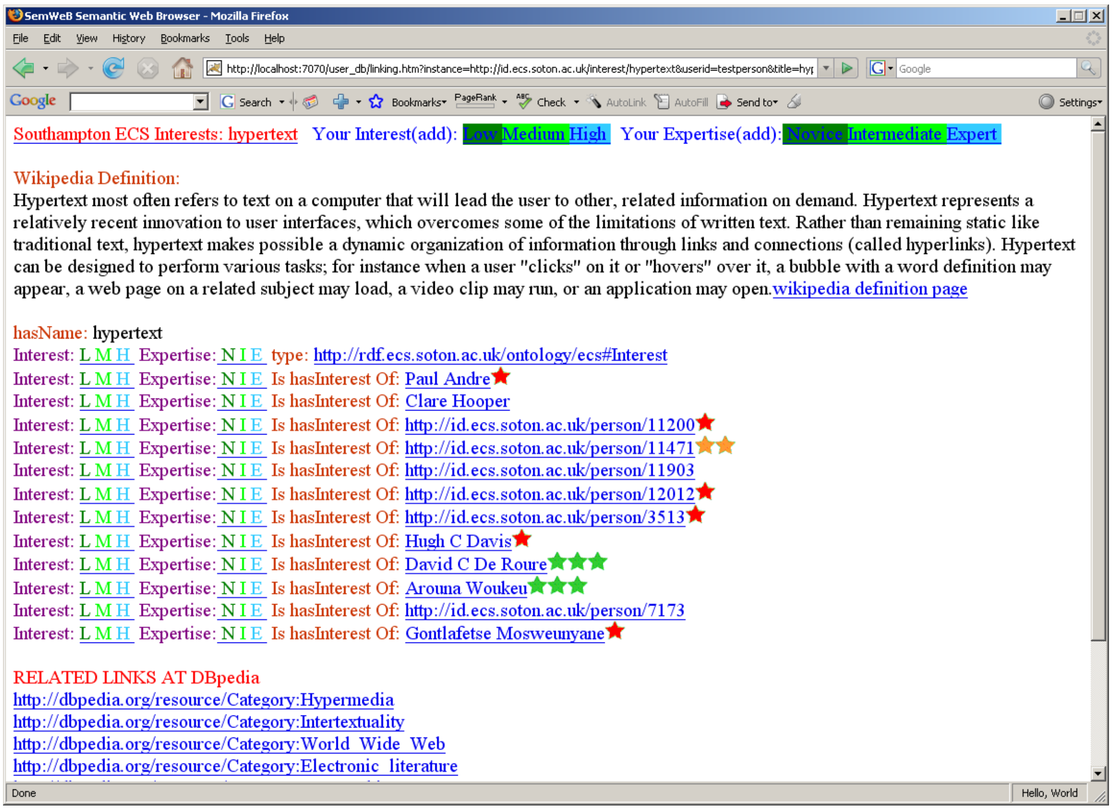


Figure D-4. Sue investigates related people and topics using SemWeB goal services and annotated links

Scenario 2: Assume Maria is an administrator in a company and she is reading daily news from the BBC news website. She is interested in an article as shown in Figure D-5. However, she couldn't find more information or related links within the page. Then, she opened the SemWeB sidebar. She annotated the Web page and requested more links as displayed in Figure D-6. From the added links, Maria followed to some of them, such as "Milky Way" galaxy (Figure D-7) and learned more information about it and followed to some hyperlinks. Then she logged in personalized contents and from the browser, she selected a browsing goal as shown in Figure D-8. From the article, then she followed to "Black Holes" and more related links to Wikipedia pages are presented at the bottom of the page (Figure D-9). Then, she added some interests to her profile (Figure D-10) and the appearance of links has changed according to her interests as shown in Figure D-11.

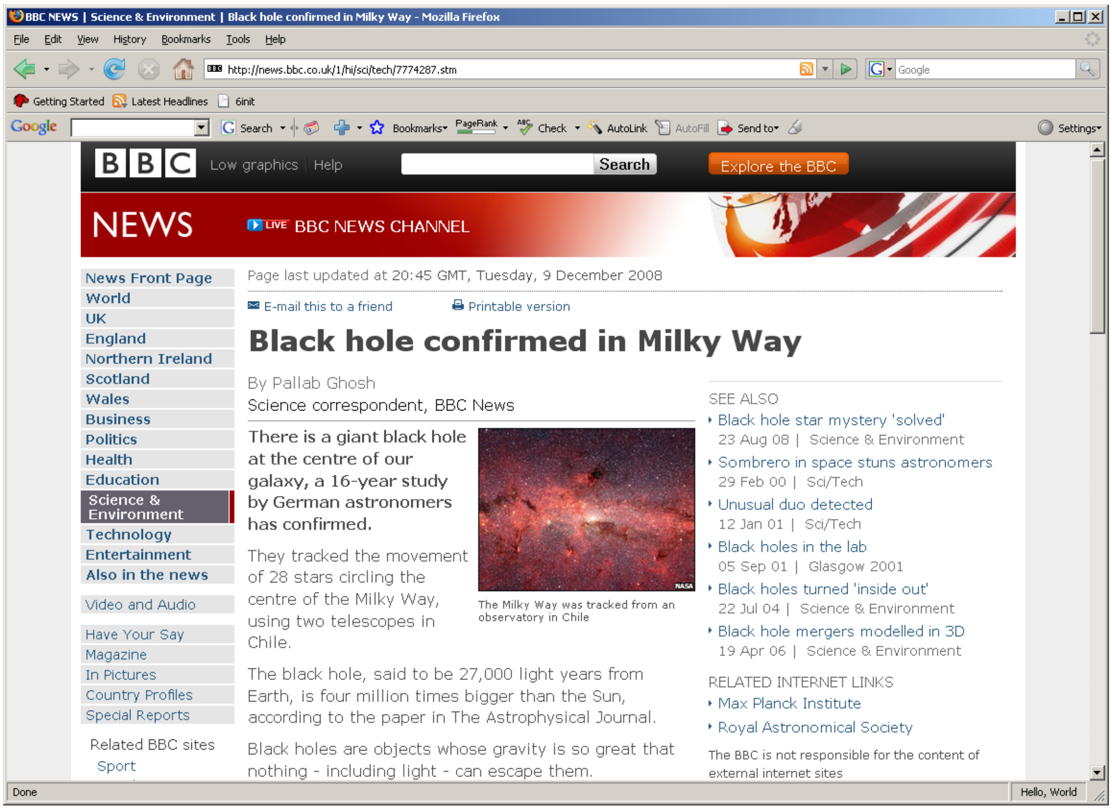


Figure D-5. An article from the BBC news Website

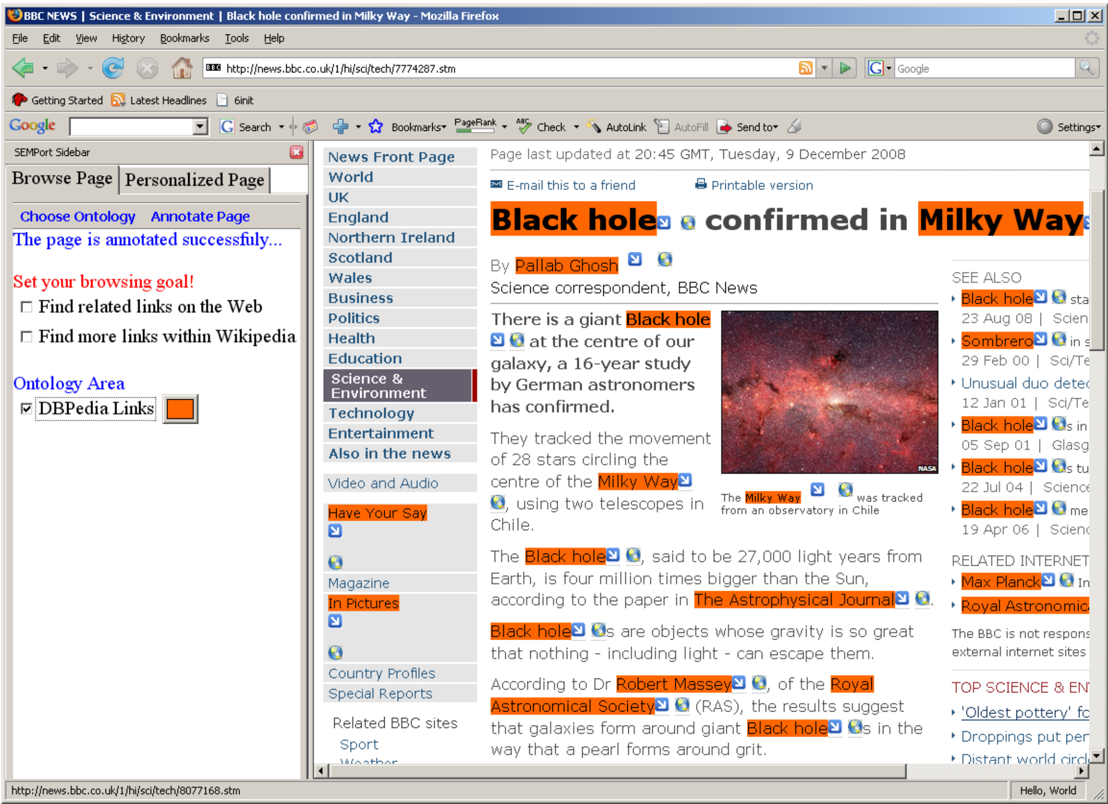


Figure D-6. The same article with the added links

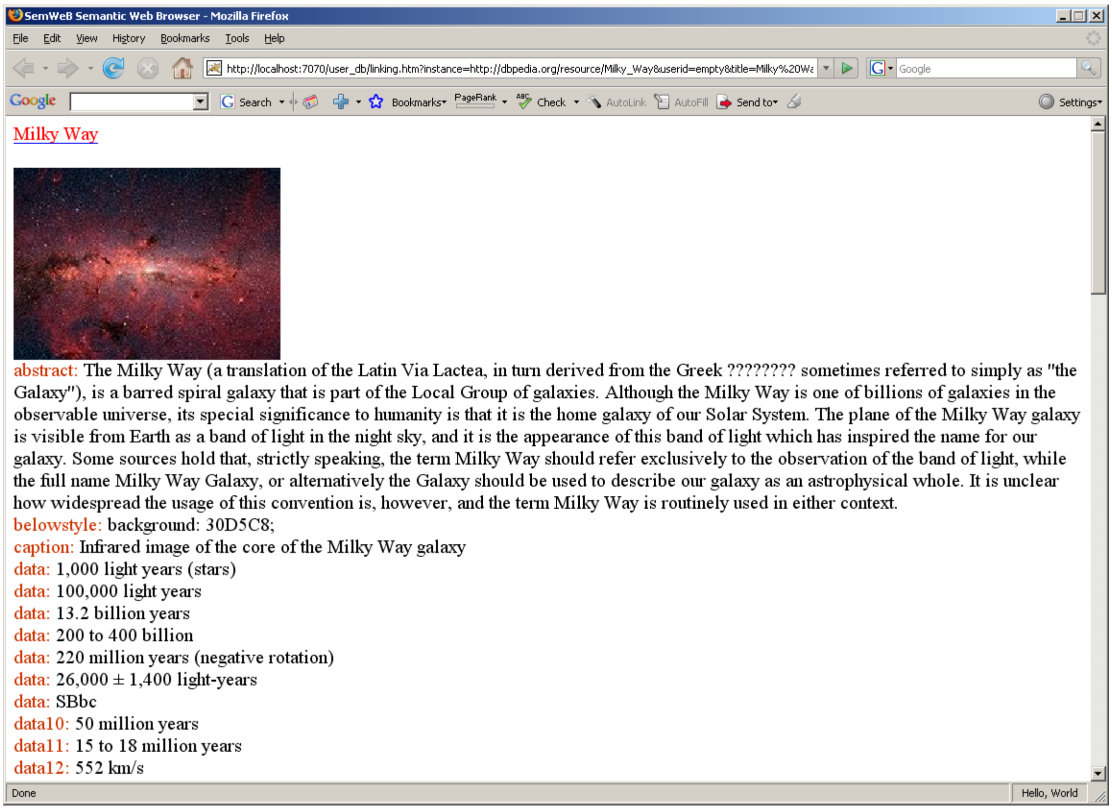


Figure D-7. Created information and links about "Milky Way"

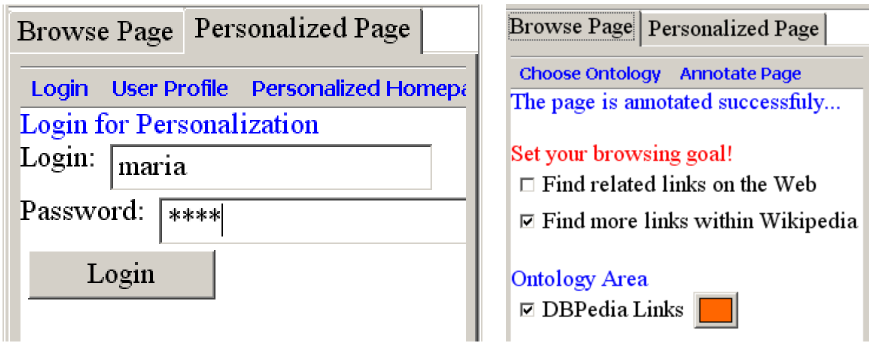


Figure D-8. Maria logged into SemWeB and selected a browsing goal

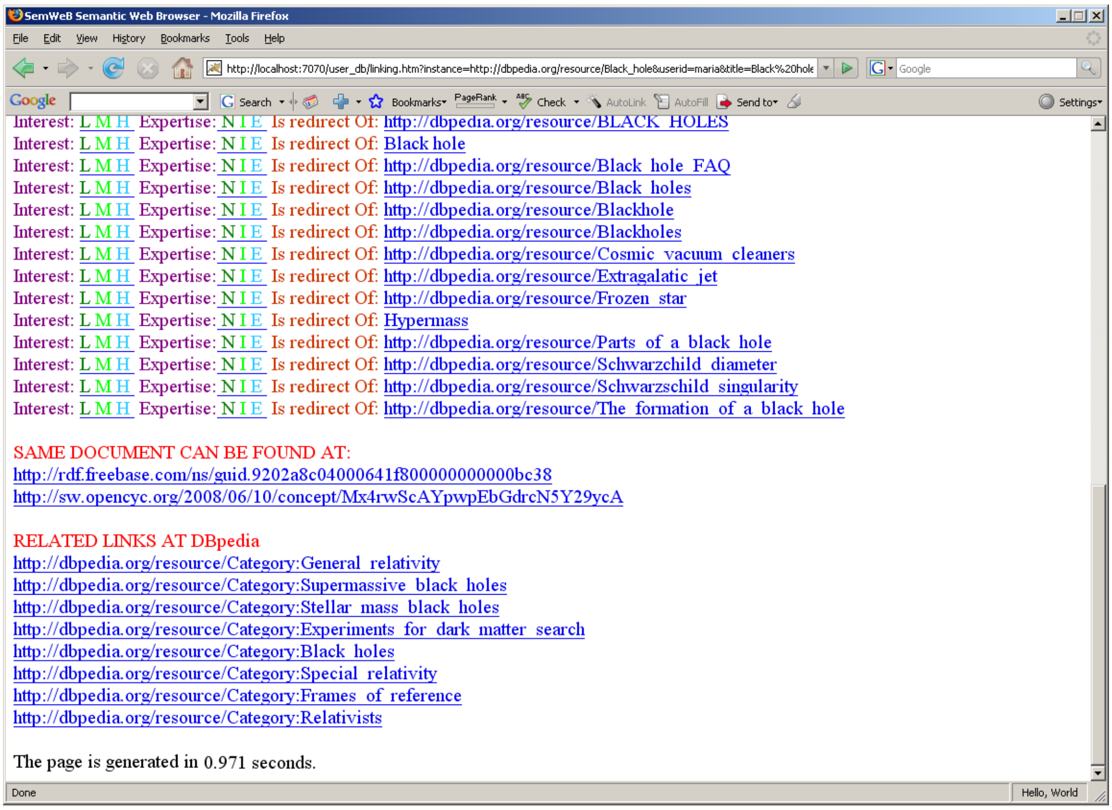


Figure D-9. Links to the related Wikipedia pages are presented based on the browsing goal at the bottom of the page

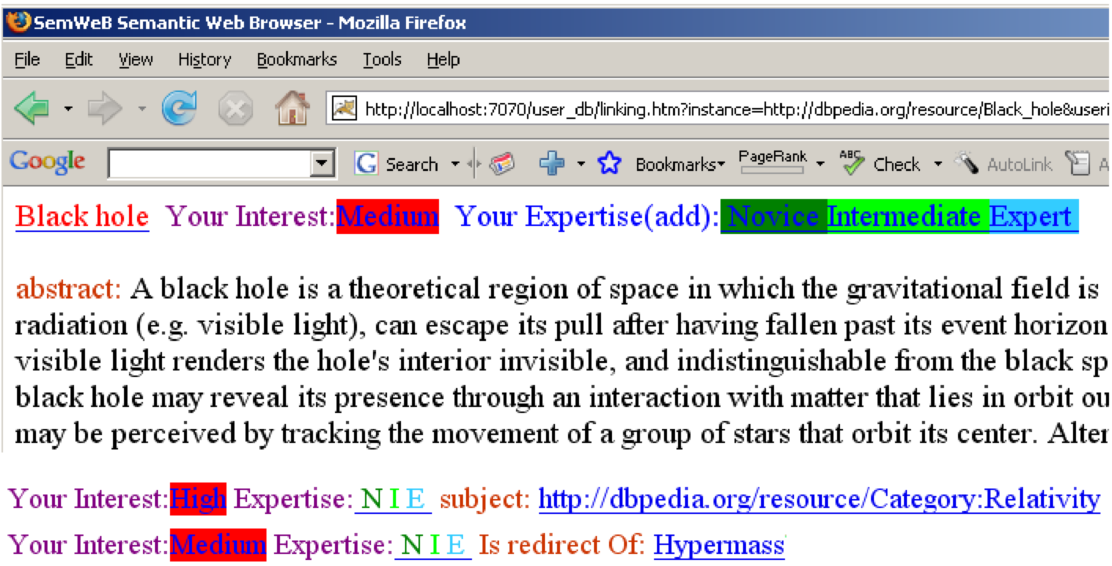


Figure D-10. Maria added interests to her profile

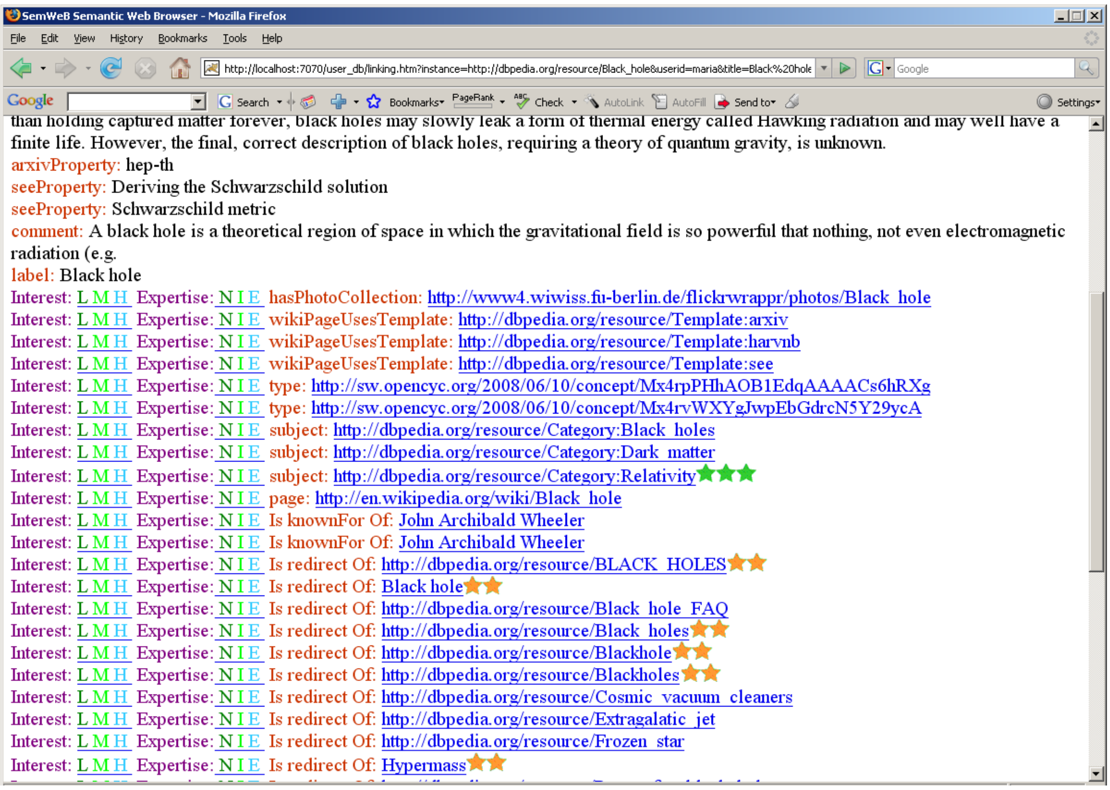


Figure D-11. Personalized links according to Maria's interests

Appendix E. User Model Ontology

RDF/XML serialization of the user model ontology, which is created by the Protégé ontology editor:


```xml
<?xml version="1.0"?>
<rdf:RDF
        xmlns:foad="http://xmlns.com/foaf/0.1/"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns="http://localhost:7070/user_db/user_schema.owl#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xml:base="http://localhost:7070/user_db/user_schema.owl">
<owl:Ontology rdf:about="">
        <rdfs:label>User Ontology (First Version - July 2007)</rdfs:label>
        <owl:imports rdf:resource="http://purl.org/dc/elements/1.1/"/>
        <dc:description xml:lang="en">This ontology describes a new user ontology
        to represent user profiles using user’s interests, browsing behaviours and
        expertise</dc:description>
        <dc:creator rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Melike Sah</dc:creator>
        <dc:title xml:lang="en">A user Ontology for the personalization</dc:title>
</owl:Ontology>
<owl:Class rdf:ID="Interest">
        <rdfs:label>Interest</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="ExpertiseValue">
        <rdfs:label>Expertise Value</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="BrowsingBehaviour">
        <rdfs:label>Browsing Behaviour</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Language">
        <rdfs:label>Language</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Person">
        <rdfs:label>Person</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Short_Term_Browsing_Goal">
        <rdfs:label>Short Term Browsing Goal</rdfs:label>
        <rdfs:subClassOf rdf:resource="BrowsingGoal"/>
</owl:Class>
<owl:Class rdf:ID="Expertise">
        <rdfs:label>Expertise</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Security">
```

```
		<rdfs:label>Security</rdfs:label>
	</owl:Class>
	<owl:Class rdf:about="#BrowsingGoal">
		<rdfs:label>Browsing Goal of the User</rdfs:label>
	      </owl:Class>
	<owl:Class rdf:ID="BrowsingInterest">
		<rdfs:subClassOf rdf:resource="#Interest"/>
		<rdfs:label>Browsing Interest</rdfs:label>
	</owl:Class>
	<owl:Class rdf:ID="Bookmark">
		<rdfs:subClassOf rdf:resource="#Interest"/>
	</owl:Class>
	<owl:Class rdf:ID="Address">
		<rdfs:label>Address</rdfs:label>
	</owl:Class>
	<owl:Class rdf:ID="BrowsingLevel">
		<rdfs:label>Level Of Browsing</rdfs:label>
	</owl:Class>
	<owl:Class rdf:ID="Identification">
		<rdfs:label>Identification</rdfs:label>
	</owl:Class>
	<owl:Class rdf:ID="Preference">
		<rdfs:label>Preference</rdfs:label>
	</owl:Class>
	<owl:Class rdf:ID="BrowsingType">
		<rdfs:label>Browsing Type</rdfs:label>
	</owl:Class>
	<owl:Class rdf:ID="Persistent_Browsing_Goal">
		<rdfs:subClassOf rdf:resource="#BrowsingGoal"/>
		<rdfs:label>Persistent Browsing Goal</rdfs:label>
	</owl:Class>
	<owl:ObjectProperty rdf:ID="HasSecurity">
		<rdfs:range rdf:resource="#Security"/>
		<rdfs:domain rdf:resource="#Person"/>
		<rdfs:label>Security</rdfs:label>
	</owl:ObjectProperty>
	<owl:ObjectProperty rdf:ID="HasBehaviour">
		<rdfs:label>Behaviour</rdfs:label>
		<rdfs:domain rdf:resource="#Person"/>
		<rdfs:range rdf:resource="#BrowsingBehaviour"/>
	</owl:ObjectProperty>
	<owl:ObjectProperty rdf:ID="HasUserType">
		<rdfs:label>User Type</rdfs:label>
	</owl:ObjectProperty>
	<owl:ObjectProperty rdf:ID="HasGoal">
		<rdfs:range rdf:resource="#BrowsingGoal"/>
		<rdfs:domain rdf:resource="#Person"/>
		<rdfs:label>Goal</rdfs:label>
	</owl:ObjectProperty>
	<owl:ObjectProperty rdf:ID="HasExpertiseValue">
		<rdfs:label>Has Expertise Value</rdfs:label>
```

```
            <rdfs:range rdf:resource="#ExpertiseValue"/>
            <rdfs:domain rdf:resource="#Expertise"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasBrowsingLevel">
            <rdfs:domain rdf:resource="#BrowsingBehaviour"/>
            <rdfs:label>Browsing Level</rdfs:label>
            <rdfs:range rdf:resource="#BrowsingLevel"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasBrowsingType">
            <rdfs:range rdf:resource="#BrowsingType"/>
            <rdfs:label>Browsing Type</rdfs:label>
            <rdfs:domain rdf:resource="#BrowsingBehaviour"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasPortfolio">
           <rdfs:label>Portfolio</rdfs:label>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasInterest">
           <rdfs:range rdf:resource="#Interest"/>
           <rdfs:label>Interest</rdfs:label>
           <rdfs:domain rdf:resource="#Person"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="LanguagePreference">
            <rdfs:range rdf:resource="#Language"/>
            <rdfs:domain rdf:resource="#Preference"/>
            <rdfs:label>Language Preference</rdfs:label>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasAddress">
            <rdfs:range rdf:resource="#Address"/>
            <rdfs:domain rdf:resource="#Identification"/>
            <rdfs:label>Address</rdfs:label>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasIdentification">
            <rdfs:label>Identification</rdfs:label>
            <rdfs:range rdf:resource="#Identification"/>
            <rdfs:domain rdf:resource="#Person"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasExpertise">
            <rdfs:domain rdf:resource="#Person"/>
            <rdfs:label>Has Expertise</rdfs:label>
            <rdfs:range rdf:resource="#Expertise"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasLanguage">
            <rdfs:label>Has Language</rdfs:label>
            <rdfs:range rdf:resource="#Language"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="HasPreference">
            <rdfs:label>Preference</rdfs:label>
            <rdfs:domain rdf:resource="#Person"/>
            <rdfs:range rdf:resource="#Preference"/>
    </owl:ObjectProperty>
    <owl:DatatypeProperty rdf:ID="GoalDescription">
```

```xml
		<rdfs:label>Goal Description</rdfs:label>
		<rdfs:domain rdf:resource="#BrowsingGoal"/>
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="City">
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
		<rdfs:label>City</rdfs:label>
		<rdfs:domain rdf:resource="#Address"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="name">
		<rdfs:label>Name</rdfs:label>
		<rdfs:domain rdf:resource="#Identification"/>
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="GoalDate">
		<rdfs:domain rdf:resource="#BrowsingGoal"/>
		<rdfs:label>Goal Date</rdfs:label>
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="Street">
		<rdfs:domain rdf:resource="#Address"/>
		<rdfs:label>Street</rdfs:label>
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="GoalType">
		<rdfs:label>Goal Type</rdfs:label>
		<rdfs:domain rdf:resource="#BrowsingGoal"/>
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="InterestDescription">
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
		<rdfs:domain rdf:resource="#Interest"/>
		<rdfs:label>Interest Description</rdfs:label>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="LayoutPreference">
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
		<rdfs:label>Layout Preference</rdfs:label>
		<rdfs:domain rdf:resource="#Preference"/>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="InterestDateCreated">
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
		<rdfs:domain rdf:resource="#Interest"/>
		<rdfs:label>Interest Date Created</rdfs:label>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="InterestDateModified">
		<rdfs:domain rdf:resource="#Interest"/>
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
		<rdfs:label>Interest Date Modified</rdfs:label>
	</owl:DatatypeProperty>
	<owl:DatatypeProperty rdf:ID="LangaugeDescription">
		<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
        &lt;rdfs:domain rdf:resource="#Language"/&gt;
        &lt;rdfs:label&gt;Language Description&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="GoalDateModified"&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/&gt;
        &lt;rdfs:label&gt;Goal Date Modified&lt;/rdfs:label&gt;
        &lt;rdfs:domain rdf:resource="#BrowsingGoal"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="Country"&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
        &lt;rdfs:label&gt;Country&lt;/rdfs:label&gt;
        &lt;rdfs:domain rdf:resource="#Address"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="email"&gt;
        &lt;rdfs:label&gt;Email&lt;/rdfs:label&gt;
        &lt;rdfs:domain rdf:resource="#Identification"/&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="AddressType"&gt;
        &lt;rdfs:domain rdf:resource="#Address"/&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >The type of the adress (i.e. home, work, etc.)&lt;/rdfs:comment&gt;
        &lt;rdfs:label&gt;Address Type&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="GoalPriority"&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
        &lt;rdfs:label&gt;Goal Priority&lt;/rdfs:label&gt;
        &lt;rdfs:domain rdf:resource="#BrowsingGoal"/&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="LanguageValue"&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
        &lt;rdfs:domain rdf:resource="#Language"/&gt;
        &lt;rdfs:label&gt;Language Value&lt;/rdfs:label&gt;
&lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="About"&gt;
        &lt;rdfs:label&gt;About&lt;/rdfs:label&gt;
        &lt;rdfs:domain&gt;
          &lt;owl:Class&gt;
            &lt;owl:unionOf rdf:parseType="Collection"&gt;
              &lt;owl:Class rdf:about="#Interest"/&gt;
              &lt;owl:Class rdf:about="#Expertise"/&gt;
            &lt;/owl:unionOf&gt;
          &lt;/owl:Class&gt;
        &lt;/rdfs:domain&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
       &lt;/owl:DatatypeProperty&gt;
&lt;owl:DatatypeProperty rdf:ID="Postcode"&gt;
        &lt;rdfs:label&gt;Postcode&lt;/rdfs:label&gt;
        &lt;rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/&gt;
        &lt;rdfs:domain rdf:resource="#Address"/&gt;
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="InterestRating">
        <rdfs:label>Interest Rating</rdfs:label>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        <rdfs:domain rdf:resource="#Interest"/>
        <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Values (1-3)</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="ColorSchemePreference">
        <rdfs:label>Color Scheme Preference</rdfs:label>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:domain rdf:resource="#Preference"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="language">
        <rdfs:domain rdf:resource="#Language"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:label>Language</rdfs:label>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="UserName">
        <rdfs:domain rdf:resource="#Security"/>
        <rdfs:label>User Name</rdfs:label>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Password">
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <rdfs:label>Password</rdfs:label>
        <rdfs:domain rdf:resource="#Security"/>
</owl:DatatypeProperty>
<BrowsingLevel rdf:ID="active">
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Active</dc:title>
</BrowsingLevel>
<ExpertiseValue rdf:ID="expert">
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Expert</dc:title>
</ExpertiseValue>
<BrowsingType rdf:ID="undirected">
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Undirected</dc:title>
</BrowsingType>
<ExpertiseValue rdf:ID="novice">
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Novice</dc:title>
</ExpertiseValue>
<BrowsingLevel rdf:ID="inactive">
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Inactive</dc:title>
</BrowsingLevel>
<BrowsingType rdf:ID="directed">
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Directed</dc:title>
```

```
</BrowsingType>
<BrowsingLevel rdf:ID="very-active">
	<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
	>Very Active</dc:title>
</BrowsingLevel>
<BrowsingLevel rdf:ID="passive">
	<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
	>Passive</dc:title>
</BrowsingLevel>
<BrowsingType rdf:ID="semi-directed">
	<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
	>Semi-directed</dc:title>
</BrowsingType>
<ExpertiseValue rdf:ID="intermediate">
	<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
	>Intermediate</dc:title>
</ExpertiseValue>
</rdf:RDF>
```