

# Building and Managing Personalized Semantic Portals

Melike Şah

Electronics and Computer Science, University of  
Southampton, SO17 1BJ, United Kingdom  
+44 (0) 23 8059 7688  
ms305r@ecs.soton.ac.uk

Wendy Hall

Electronics and Computer Science, University of  
Southampton, SO17 1BJ, United Kingdom  
+44 (0) 23 8059 3255  
wh@ecs.soton.ac.uk

## ABSTRACT

This paper presents a semantic portal, SEMPort, which provides better user support with personalized views, semantic navigation, ontology-based search and three different kinds of semantic hyperlinks. Distributed content editing and provision is supplied for the maintenance of the contents in real-time. As a case study, SEMPort is tested on the Course Modules Web Page (CMWP) of the School of Electronics and Computer Science (ECS).

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Retrieval models

## General Terms

Design, Experimentation, Human Factors

## Keywords

Semantic Web, Semantic Portal, Ontologies, Content Editing and Provision, Personalization

## 1. INTRODUCTION

The semantic portal refers to organized web sites that contain collections of semantically structured information. Ontologies are used for the structuring, accessing, sharing and presentation of the knowledge. The process of content provision in semantic portals is difficult, especially where content is continuously changing. This problem has been studied by many semantic portals, such as OntoWeb [1] and ODESeW [2]. For instance, in OntoWeb, new added information can not be seen without restarting the server and in ODESeW, Web interface is difficult for the edition and the insertion of relationship values. In addition, search and navigation are commonly used access mechanisms in semantic portals. However, in most of the cases, navigation is not very effective; the same page content and the same hyperlinks are shown to all users, and hyperlinks to similar pages are often not presented.

To address these problems, we propose an ontology-based semantic portal, SEMPort. We have developed an easy to use Web interface for inserting, updating and removing instances and their attributes in real-time. In addition, ontology-based user models are utilized to support personalized views. To improve the browsing facilities of users, we provide semantic navigation and ontology-based search with semantic hyperlinks (explicit, implicit and recommendation links) to relevant pages. For the illustration of SEMPort, we use CMWP of ECS.

## 2. SEMPort ARCHITECTURE

The backbone of the SEMPort architecture is a knowledge base (KB), where the domain model ontologies/instances are stored (Figure 1). The KB is connected to the Protégé ontology editor [3], which allows the update of ontologies/instances. We have also developed a Web front-end, so that any number of ontology/instance files can be uploaded to the KB. SEMPort also supports content editing and provision through its Web interface. For inferencing, the Jena Rule-based reasoner and OWL reasoner are utilized and the Jena API [4] is used for handling user and domain models. At the front-end, SEMPort is a set of Java servlets, which query the KB over HTTP using SPARQL queries. In addition, SEMPort supports reusability; ontologies, instances and rules can be easily changed.

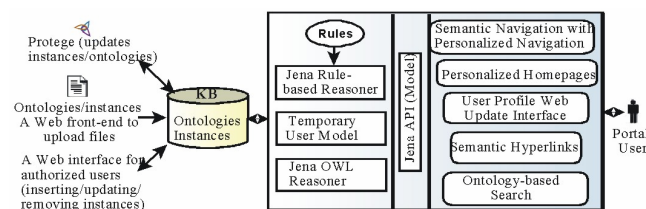


Figure 1. The Architecture of SEMPort

## 3. A CASE STUDY OF SEMPort ON CMWP

### 3.1 Domain Ontologies

The ECS CMWP consists of the list of modules, which are taught in the school. Each module has a list of topics that are covered by the module. To annotate CMWP, we used two existing ontologies; a part of the ECS ontology [5], which we call ECS\_COURSE, and the ACM Computer Classification System (CSS) [6]. We used ECS\_COURSE to manually annotate CMWP in OWL Lite, using Protégé. We also converted ACM CSS to RDF by Perl Scripts and used at the annotation of covered topics.

### 3.2 User Model

#### 3.2.1 User Ontology

To adapt to the different browsing needs of ECS CMWP users, we used a user ontology, which we call USER. USER describes different relationships between the user model and domain model concepts; *hasInterest*, *memberOf* and *hasDegree* are some relationships that are used. For example, we use the *hasInterest* relationship to indicate the user's interest on ACM CSS topics.

#### 3.2.2 User Profile

In SEMPort, personalization starts when the user log on to the portal with their ECS IDs and passwords. In the introduction, we ask a few questions to ascertain the user's background and interests. Alternatively, users can extend their profiles with semantic bookmarks, e.g. users can indicate their interest in ACM CSS topics and can associate different weights (low, medium,

high). Then, the user profiles are converted to RDF using the USER ontology. Profiles are handled as temporary user models at run-time and also stored to the client's computer as an RDF file.

### 3.3 Functionalities of SEMPport

#### 3.3.1 Semantic Navigation & Ontology-based Search

Ontologies are used for the navigation as shown in Figure 3, left pane. Users can access concept instances by clicking onto hyperlinks from the ontology hierarchy; all instances and sub-instances are queried from the KB and displayed in the right pane. When the user clicks on an instance from the right pane, a more detailed view is opened, which contains all attributes of the instance and semantic hyperlinks to relevant pages (Figure 2).



Figure 2. A view from semantic navigation

To facilitate information access, ontology-based search is integrated to semantic navigation (Figure 2). When a concept is followed from the ontology hierarchy, a search form is automatically generated; all properties and inherited properties of the concept are listed. For relationship properties, all range values (including inherited) are filled into a drop-down list, thus users can select valid relationship values easily. Users can also put constraints on numeric values (i.e. >, <). Then, the entered data is converted to SPARQL and results are interrogated from the KB.

#### 3.3.2 Semantic Hyperlinks

To guide users to relevant pages, explicit, implicit and recommendation links are provided. Explicit links reveal relationships to directly associated resources, such as relations defined by the ontology. Implicit links use explicitly defined rules to infer incomplete knowledge from the KB. Recommendation links also use rules to present interesting relations between different concepts and channel users to relevant pages (Figure 2).

#### 3.3.3 Personalization based on User Ontology

Personalization is achieved by modeling user profile as a set of concepts that has relationships to domain model concepts.

First, a Web interface is developed for users to maintain their profiles. Second, SEMPport supports personalized navigation by performing semantic weighted-ranking; we use user's interests in ACM CSS topics (bookmarks), provided weight by the user and semantics between ACM CSS for reordering the contents (Figure 2). Third, personalized homepages guide users to relevant pages. In Figure 3, we illustrate examples of student (left) and teacher (right) homepages. Based on different user types, different contents and hyperlinks are shown. If the teacher is a module leader, a link is supplied to update the contents of the module.

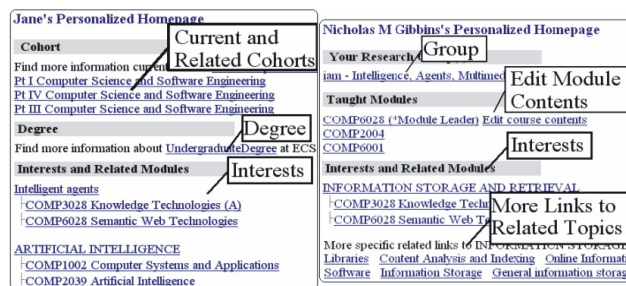


Figure 3. Personalized homepages (left-student, right-teacher)

#### 3.3.4 Content Editing/Provision on CMWP

In order to keep information up-to-date on ECS CMWP, we have developed a distributed Web interface for content editing and provision by authorized users (i.e. module leaders). Authorized users can access instances from their personalized homepages and all attributes (including superclasses) are listed for update; users can add, delete or change attributes from this interface. To simplify the insertion and update of relation instances, valid instance values (including inherited) are added into a drop-down list (Figure 4, left), so users do not have to know the details of the system. Multiple values can be added to the same attribute and multiple attributes can be deleted at the same time. Changes are permanently saved to the KB and can be seen without restarting the server. Consistency checking is also performed and presented to prevent inconsistencies (Figure 4, right).

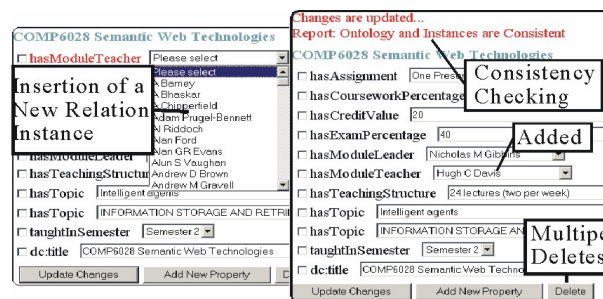


Figure 4. Web interface for content editing/provision

## 4. CONCLUSIONS AND FUTURE WORK

In this paper, we present a semantic portal, SEMPport, which provides an easy to use distributed Web interface for content editing/provision in real-time, also alleviating problems associated with navigation through personalized views, semantic hyperlinks, semantic navigation and ontology-based search. For the future work, user studies are going to be conducted.

## 5. REFERENCES

- [1] P. Spyns, D. Oberle, R. Volz, J. Zheng, M. Jarrar, Y. Sure, R. Studer, R. Meersman, OntoWeb - A Semantic Web Community Portal, PAKM2002, LNCS (2569) 2002
- [2] O. Corcho, A. Gomez-Perez, A. Lopez-Cima, V. Lopez-Garcia, M. Suarez-Figueroa, ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets, ISWC2003, LNCS, (2870) 2003
- [3] Protégé Ontology Editor, <http://protege.stanford.edu/>
- [4] Jena-Semantic Web Framework, <http://jena.sourceforge.net/>
- [5] ECS Ontology, <http://rdf.ecs.soton.ac.uk/ontology/ecss>
- [6] ACM CSS, <http://www.acm.org/class/1998/ccs98.txt>