Developing Semantic Portals

Brooke Abrahams
Victoria University, Australia

INTRODUCTION

A semantic portal is a type of community information portal that exploits semantic Web standards (Berners-Lee, Hendler, & Lassila, 2001) to improve structure, extensibility, customization, and sustainability. They are similar to a traditional cyberspace portal, except that Web resources are indexed using a rich domain ontology (a specification of key domain concepts) as opposed to, for example, a list of keywords, and are based on new Web markup languages such as Resource Description Framework (RDF) (Manola & Miller, 2004) and Ontology Web Language (OWL) (McGuinness & Harmelen, 2004). RDF provides a flexible and extensible format for describing information items and associated metadata, while OWL supports explicit representation of the domain ontologies used to classify and structure the items. Together, these enable a more decentralized approach to portal architectures. This chapter discusses comprehensive, ontology-based approaches for building high-value semantic portals. State of the art development tools and techniques are first presented both from a client-side and server-side perspective. Next, widely used methodologies and tools for building ontologies are discussed. Finally, a tool called Ontoviews is demonstrated, which has been designed to assist semantic portal developers by providing accessibility to search and dynamic linking services.

CLIENT-SIDE DEVELOPMENT

The first stage in the information item life cycle in a semantic portal is the creation of information. An information item is generally created as a conceptual instance of an ontology class using an ontology based annotator such as Cohse[1], OntoMat[2], or Shoe Knowledge Annotator[3]. These applications allow the information provider to create RDF markups, and then associate the markup to a Web page. At this stage there is still no one standard method for associating RDF with HTML. Popular annotation methods include:

- **Imbedding RDF in HTML:** This involves placing the RDF markup somewhere that it can be readily extracted while not displayed by the browser. This may be done using the head tags or comment tags of the HTML document.
- **Linking to external document:** This is arguably the purest solution from an architectural point of view. The RDF annotations are stored on a separate RDF file somewhere on the Web. The original HTML source document then contains a `<link>` to the annotation. One drawback of this method is that maintaining the metadata externally to the RDF source document can be an inconvenience.
- **Embed RDF as XHTML:** This approach basically involves hacking up a small DTD (document type definition) using XHTML Modularization for a variant of XHTML, putting it on the Web, and then referencing it from the source document. The main drawback with this method is that the DTDs are large and relatively complex; this is not a viable approach for typical HTML authors.

The most commonly used approach to annotation, however, is to embed the markup in the head or comment tags of an HTML file, as shown in Figure 1. The information can then be extracted by a Web crawling application and mediated with the ontology schema.

SERVER-SIDE DEVELOPMENT

Semantic portals require a means to store information in an RDF enabled database, retrieve documents from the database, process RDF statements to infer knowledge, aggregate information from different sources, including other domains, and process RDF queries. Semantic middleware applications facilitate the above tasks by providing a platform with access to required functionality. Developers can access pre-existing modules for storing, retrieving, querying, and inferring knowledge, by interfacing with a middleware environment via an application programming interface (API). Table 1 is a list of some of the most popular middleware environments. For some time the leading framework has been Jena[4].

The middleware environments in Table 1 provide access to a type of program called a reasoner. Reasoners can be employed to check cardinality constraints and class membership, or infer new knowledge from existing knowledge based on the semantics specified in an ontology. Examples of description logic reasoners are Racer Pro[5], Pellet[6], and Fact[7]. The environments also contain a query engine for processing RDF queries. Work on RDF query languages

Figure 1. Annotated Web site (Abrahams & Dai, 2005)

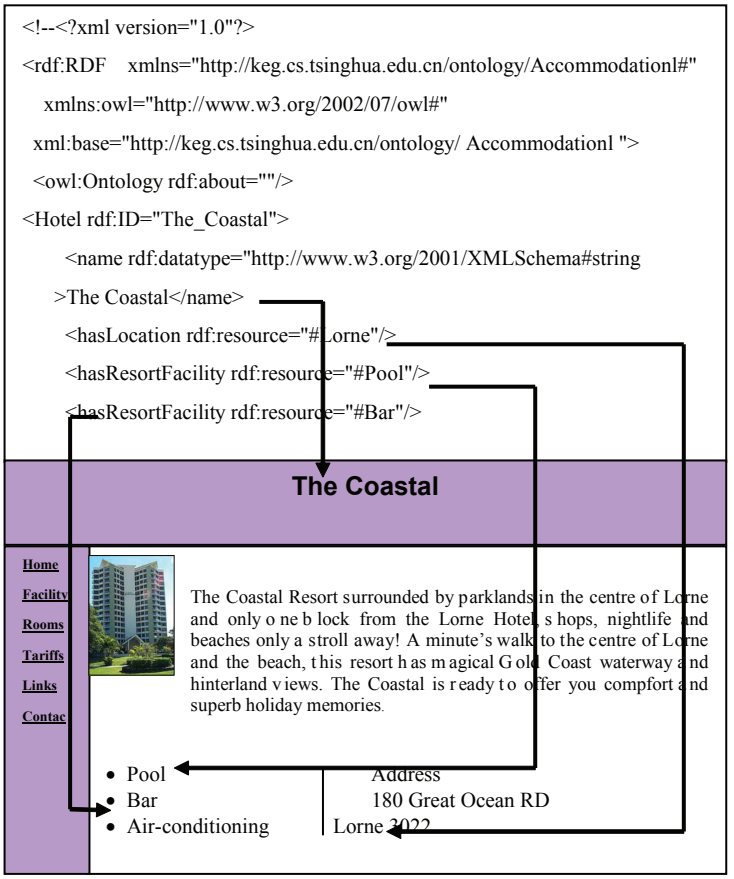

Table 1. Semantic middleware environments

Developer	**Product**	**Category**
Administrator *http://www.aidmistotor.nl/*	Sesame Spectacle	RDF(S) storage and retrieval, ontology-based information presentation
FZI – AIFB *http://kaon.semanticweb.org/frontpage*	KAON	Inference engine, knowledge management, and tools
HP Labs *http://jena.sourceforge.net/*	Jena	Inference engine, knowledge management, and tools
Intellidimension *http://www.intellidimension.com/*	RDF Gateway	RDF data management system
Kowari *http://www.kowari.org/*	Kowari Metastore	Metadata analysis and knowledge discovery, RDF storage
Ontoprise *http://www/ontoprise.de/*	Ontobroker	Inference middleware