Meeting Semantic Web Challenges with Automated Annotation and Multiagent Querying of Web Resources

Brooke Abrahams and Wei Dai
School of Information Systems Victoria University

Abstract
The goal of the Semantic Web initiative is to enable machine understanding of Web resources, thus providing the foundation for semantic architecture to support transparent exchange of information and knowledge among collaborating e-business organizations (Singh et al, 2005). Major challenges, however, remain for developers of semantic Web applications before the technology can be considered a viable mainstream part of the Internet. The challenges include i) improving the availability of semantic Web content (Benjamins et al, 2004), and ii) providing applications that facilitate effective ontology based information retrieval (Chebotko et al, 2004). This paper presents a framework for addressing these issues. The architecture is designed to offer unprecedented opportunities for precise querying and searching of Web resources. An easy to use annotation tool is deployed, providing a convenient mechanism for Web site owners to mark up their Web pages with RDF metadata. Search and coordination activities are carried out by a system of multiagents based on the index environment outlined in (Dai and Wright, 1996). To demonstrate proof of concept, the architecture is applied to the Accommodation services domain of the Australian Tourism Industry.

Keywords
Semantic Web, RDF, OWL, Ontology, Annotation, Multiagent Querying.

Introduction
When surfing for Internet services, end users are increasingly in need of more powerful tools capable of collecting and interpreting the vast amount of heterogeneous information available on the Web. This heterogeneity stems from the fundamental disparity of Web domains. In the tourism industry for example, there is vast amounts of information available about accommodation, transportation, entertainment, and insurance. The information has severe limitations, however, because it is largely displayed in HTML, which is designed for humans to read rather than machines to interpret and automatically process. What is needed is a system based on global schemas where information can be interpreted and exchanged by machines. The semantic Web is an initiative by the W3C in a collaborative effort with a number of scientists with the goal of providing machine readable Web intelligence that would come from hyperlinked vocabularies, enabling Web authors to explicitly define their words and concepts. The idea allows software agents to analyze the Web on our behalf, making smart inferences that go beyond the simple linguistic analysis performed by today’s search engines (Alesso and Smith, 2004). The applications that deliver these new online solutions are based on ontologies. An ontology is basically a description of the key concepts in a given domain including the rules, properties and relationships between concepts. There are many challenges involved in implementing such an innovative new approach for online search services. Availability of semantic Web content (Benjamins et al, 2004) and ontology based information retrieval (Chebotko et al, 2004) are two of the major issues faced by developers. In this paper, an architecture designed to address these issues is presented. The aim is to i) provide convenient and effective ways for users to annotate Web pages, thus facilitating wider availability of semantic Web content, and ii) deploy effective applications, based on a multiagent system to facilitate concept based querying of semantic Web resources. To demonstrate the proof of concept, the architecture will be applied to accommodation services in the Australian Tourism Industry.
Theoretical Framework

Semantic Web Markup Languages

In recent years, new Web markup languages have begun to emerge allowing data on the Web to be defined and linked in ways that are more effective for discovery, automation, integration, and reuse across different applications. The RDF (Resource Description Framework) and OWL (Ontology Web Language) are two of the major initiatives. RDF provides a simple way for descriptions to be made about Web resources using a set of triples based on description logic. RDF is limited to descriptions about individual resources and does not provide any modeling primitives for the development of ontologies. RDFS extends RDF by providing a vocabulary by which we can express classes, their subclass relationships as well as define properties and associate them with classes. OWL builds on RDFS to provide more vocabulary for defining complex relationships between classes like disjointness, cardinality of properties, and richer semantic capability such as symmetry. As a result of this expressive power, semantic Web languages are able to facilitate inference and enhanced searching of Web content. Suppose for example, we have information about 3 people, Bill, Bob and Jane, and have described the relationship between these 3 people using an OWL ontology. If we knew that that Bill has an ancestor called Bob, and Bob has an ancestor called Jane, by making the ‘hasAncestor’ property transitive, we could infer that Bill also has and ancestor called Jane, even though the relationship between Bob and Jane is not explicitly stated. Likewise, if the ‘hasAncestor’ property was also made inverse, we could infer that Bob is the ancestor of Bill, Jane is the ancestor of Bob, and Jane is also the ancestor of Bill.

![Fig.1. Asserted and Inferred Relationships](image)

Semantic Web Architecture

Fig 2 represents a framework to support convenient annotation and intelligent querying of semantic Web resources. Annotation software is used by a Web site owner to generate RDF markup describing the content of their Web site. The RDF markup is essentially instance data that conforms to an OWL domain ontology. Web agent software crawls the Internet at regular intervals searching and extracting RDF marked up documents consistent with the domain ontology. The extracted RDF content is then stored in an RDF enabled database which forms part of a semantic middleware application maintained on a Web server. The Web agents also have access to the domain ontology and store a local copy in the RDF database. The GUI is accessed remotely by an end user searching for Web content in the same way as a conventional search engine. The user requests are passed to the Web agents which in turn formulate a query plan. Inferencing is performed on ontology schema information and instance data by the activation of a reasoner which is a component of the middleware. Queries are initiated in the middleware application by the agents and results displayed to the end user via the GUI.

Victoria University Business Research Conference 2005
Note: In the following sections of this paper, the Fig 2 framework will be demonstrated in the context of Accommodation services in the Australian Tourism Industry.

Web Site Annotation

Annotation Software Usability Requirements

Currently there is little semantic Web content available on the Internet (Benjamins et al, 2004). Part of the problem is that manual annotation of content with metadata is not scaleable, and there is a lack of tools capable of annotating documents automatically (Alesso and Smith, 2004). For semantic Web content to become widely available there needs to be integration of domain ontologies and automated annotation tools retrieval (Chebotko et al, 2004). In February 2005, a survey was conducted (as part of this research) of 384 businesses listed with AAA Tourism (Australia’s official online guide to star rated accommodation). The aim was to find out, for the purpose of assisting in the design of an annotation tool for the accommodation domain, i) how likely Web site owners were to use a new Internet technology ii) what factors would encourage them to do so, and iii) how they would like a new Internet technology to be applied to their business. Survey results found that 342 out of 384 businesses, a figure of 89% answered either maybe, likely or definitely to the question of rebuilding or overhauling their website in order to use a new Internet technology. Encouragingly 52% of businesses said they were likely or would definitely overhaul their Website to use a new technology, which may suggest a positive attitude towards the adoption of the semantic Web. To the question of “what factors would influence businesses to overhaul their Web site in order to use a new Internet technology?” (multiple answers were permitted for the question), the most popular factor was “if it was proven to increase Web exposure”, with 80% of respondents choosing this option. 78% selected “If it was easy to use”, followed by “The cost of implementing it was low”, which was chosen by 77% of those surveyed. In addition, 61% indicated that they would prefer the technology to be added to their existing Web site, rather than having to build a new Web site in order to use it. These results appear to suggest that the best chance of getting businesses to adopt the semantic Web is for i) the benefits of it to be well communicated to potential users, ii) make sure the technology is easy to apply, and iii) ensure the cost of implementation is low.

An Annotated Website Generator

AcOntoWeb is a software application that allows accommodation Web site owners to automatically annotate their Web sites in accordance with a domain ontology. The software was designed by taking into consideration of the usability requirements that accommodation providers have for a new Internet technology, as discussed in 3.1 (i.e. the software is easy to use, allows annotations to be added to existing Web sites, and will be made available as freeware).
The user is initially prompted to enter the resort contact details and select a location.

They then enter a description of the resort and select a picture from file (if generating a new Web site) that will appear on the main page of their Web site.

The next step is to select resort and room facilities that are offered by the business from a list of checkboxes.

The user is then prompted to select a star rating as well as given the option to either generate a new Web site, or to annotate an existing Web site. If they choose to annotate an existing Web site, they are prompted to open the home page of their Web site and the RDF annotation will be imbedded and saved between the head tags. If the user chooses to generate a new Web site, a new annotated site will be created with link pages (e.g. Facilities, Location etc), based on selections made using AcOntoWeb. An example AcOntoWeb generated Web page and its associated RDF annotations are shown in Fig 6.
Abraha ms and Dai

The RDF annotations start with namespace declarations. Namespaces act like a prefix to associate individual resources with a particular schema. They also define the markup languages used in the ontology. Conceptually, the annotations can be thought of as instance data of the domain ontology. Fig 7 shows the relationship between ontology schema information for accommodation and instances relating to the Coastal Hotel. The class ‘Resort’ is related the class ‘Location’ which is related to the class attraction. The ‘Resort’ class is also related to the ‘Rating’ class and the ‘Facilities’ class which has 2 subclasses, ‘Room Facilities’ and ‘Resort Facilities’. The ‘Coastal Hotel’ is now an instance of the ‘Resort’ class and has the asserted location ‘Lorne’ which is an instance of the Location class. It also has the asserted rating of ‘5 Star’ which is an instance of the ‘Rating’ class. It has an asserted room facility ‘Air-conditioning’ and asserted resort facilities ‘Bar’ and ‘Pool’.

The Coasta l Resort surrounded by parklands in the centre of Lorne and only one block from the Lorne Hotel, shops, nightlife and beaches only a stroll away! A minute’s walk to the centre of Lorne and the beach, this resort has magical Gold Coast waterway and hinterland views. The Coastal is ready to offer you comfort and superb holiday memories.

Fig.6. Annotated Web Page

Fig.7. Accommodation Ontology (Partial)

Reasoning over Annotated Web Sites

A reasoner (also known as an inference engine) is an application that can read an ontology model with its schema and instance data, and derive additional knowledge about the domain. By applying a reasoner to the accommodation ontology extra facilities and attractions, other than those explicitly stated on the Web page, can be inferred as being associated with a particular resort. Fig 8 shows the...
asserted along with the inferred attractions and facilities of the Coastal Hotel. Inferred entities are depicted in the diagram using a broken line.

With the use of class restrictions, the ontology shown in Fig 5 specifies that if a resort is rated 5 Star it must have a restaurant, pay TV and valet parking. It can therefore be inferred that because the Coastal is a 5 Star resort, it must have these facilities. It can also be inferred by the use of transitive properties that because the hotel’s location is Lorne, and that Lorne has the attractions of surfing and hiking, that surfing and hiking must also associated with the Coastal.

**Multiagent Querying of Semantic Web Resources**

**Multiagent Architecture**

Agents are programs that collect content from diverse sources and exchange the results with other programs (Singh et al, 2005). They have the ability to act autonomously and proactively in order to accomplish tasks and process requests. In a multiagent system, knowledge action and control of tasks are distributed among agents which may cooperate, compete or coexist depending on the context (Houssein et al, 2004). A lot of research has been done recently in the area of multiagent coordination in a semantic Web environment. Work undertaken includes areas such as the use of agents for Web services as described in (Huhns, 2002), and multiagent infrastructures for developing Web based systems (Sycara, 2003). With these types of systems, Web search agents do not perform like commercial search engines, which use database lookups from a stored knowledge base. Instead, the agents search the Web itself, providing an interface to the user (Alesso, 2004). For this research a set of multiaagents is used that have been generated from the INDEX environment described in (Dai and Wright, 1996). Agents in this environment, unlike the systems described in (Huhns, 2002), and (Sycara, 2003), work with a semantic knowledge base contained in the Jena middleware. The agents plan and execute interaction with Jena in order to satisfy user requests. The individual agent activities are coordinated within a Java based environment in which activities are taking place concurrently, i.e. agents are carrying out their designated routine tasks. The general multiagent coordination infrastructure supported by the INDEX environment is illustrated in Fig 9. The actual implementation of the framework only includes the Accommodation domain to this point in time. The ‘Car Hire’ and ‘Airline’ domains are included in Fig 9 to show how that architecture is designed to allow other domains to be added at a later time.
The numbers shown in fig 9 correspond to the key processes below:

1. Coordination agent instructs domain agents to crawl the Internet to update domain ontologies and search for RDF annotated Web sites.

2. Domain agents search for and download relevant domain ontologies from the Web.

3. The ontologies are sent by the domain agents to the Jena agent whom is responsible for interacting with the Jena middleware application.

4. Having established a connection to the Jena middleware, the Jena agent creates a Jena ontology model and saves the model using Jena’s persistent storage capability linked to a backend database.

5. Domain agents crawl the Internet searching for and downloading Web pages with RDF markup containing a matching namespace to their domain specific ontology.

6. Domain agents extract the RDF annotations from the Web pages and send them to the Jena agent.

7. The Jena agent, having maintained a connection to Jena middleware writes the extracted RDF markup into the relevant ontology model contained in the persistent storage database.

8. End user issues requests for a travel service via the GUI.

9. GUI accepts the user request, converts the request to an XML form and sends it to the interface agent.

10. Interface agent receives the user request and transforms the task descriptions into technical specifications which are then are passed to the Coordination agent.

11. Coordination agent divides tasks into subtasks, formulates a plan and allocates subtasks tasks to domain agents.

12. Domain agents formulate a number of possible solutions to their specific tasks and convert the solutions into query specifications. The query specifications are each given a ranking based on best match to user request. Specifications are then sent to Jena agent.

13. Jena agent converts the query specifications into RDQL query language format using parameters and predefined query templates. Jena agent also invokes the Racer reasoner to classify the
ontology models which now contain both schema and instance data for each domain. Jena agent then initiates RDQL queries over the inferred ontology model.

14. Jena agent retrieves the query results from the reasoner.

15. Results are sent back to the domain agents.

16. Domain agents sort the query results into their ordered hierarchy and send them to coordination agent.

17. Coordination agent confirms that a solution has been found. It determines how results are to be displayed (order and number of hits etc) and sends the requirements and results to the interface agent.

18. Interface agent converts the results to HTML, formulates a page layout and passes results to the GUI.

19. GUI displays the results to the end user.

**Accommodation Query Example**

In this section the querying of AcOntoWeb annotated Web sites will be demonstrated. For the purpose of conducting the following experiment, a large number of Web sites were annotated using fictitious data and placed on a Web server:

A tourism customer issues a query selecting a 5 Star Hotel/Motel with a swimming pool, bar, restaurant and valet parking. Room facilities are to include pay TV and air-conditioning (see Fig 10). The attractions hiking and surfing have also been selected in the search criteria. The customer is flexible about the exact location of the resort so has left the Location check box blank. Victoria (Australia) is the preferred state. Once the user presses submit, the query is processed by the agents.

![Fig.10. GUI (query interface)](image)

The above query has returned a list of matching results shown in Fig 11. The results are displayed in an ordered of hierarchy of closest match to the user request.

![Fig.11. Query Results](image)

The results have been arrived at by querying an expanded version (now including all resorts, facilities, and attractions) of the inference model partly shown in Fig 8 of section 3.2. The Coastal Hotel is
Ontology Merging and Alignment

So far the research has only been applied to Web sites annotated according to the accommodation ontology presented in Fig 7. It is not a practical in the real world to assume that all data would be market up consistent with a single domain ontology. In fact there are many ways in which a domain could be modeled. One of the major advantages of the semantic Web is that ontologies make it possible to merge domain knowledge into a single model, thus facilitating federated queries across multiple data sources. There are many documented techniques in the literature for ontology merging, including Onions, FCA-Merge, and the Prompt (Gomez-Perez, el at, 2004). These techniques all generate a new unique ontology model from the original models. Jena has built in mechanisms for handling ontology merging using imports. Because the architecture presented in this paper uses Jena as a middleware layer, it seems practical to use the Jena approach to merging. Fig 12 shows how ontology imports are handled by Jena. Ontology A imports B so that a new unified model is created containing both sets of ontology constructs. The original models are maintained separately so that Jena still knows where statements originate from.

Once the ontologies have been merged into a single model, individual classes and properties need to be aligned. Fig 13 shows the concept ‘Resort’ with its associated properties in domain A, and a similar concept called ‘Hotel’ with its associated properties in domain B. Notice that not all properties in fig 13 are not able to be aligned. Bill code in domain A has no equivalent property in Domain B. One possible solution for overcoming this would be to map Bill Code to a dummy attribute. Alternatively, if Bill Code is not a required field in domain A, the mapping may simply be left out.

OWL provides vocabulary to align classes and properties to each other. Two classes may be stated to be equivalent by use of the equivalence axiom. The example below declares that the concepts ‘Resort’ and ‘Hotel’ are equivalent. For example:

```xml
<owl:Class rdf:ID="Resort">
  <owl:equivalentClass rdf:resource="Hotel"/>
</owl:Class>
```

Properties are declared equivalent in the same manner. The properties ‘Name’ and ‘Title’ also need to be declared as equivalent. For example:
This form of integration, however, still requires that data be annotated with RDF markup consistent with an ontology from some domain. The reality is that most of the data on the Internet today is in HTML and this is likely to remain the case for a number of years. In the mean time, other forms of integration that can capture and process non annotated data need to be considered. Latent Semantic Indexing (LSI) (Alesso, 2004) is one particular information retrieval method that organizes existing information into a semantic structure. The application of techniques such as LSI is noted as import to this area of research, but falls outside the scope of this paper.

**Conclusion**

The evolution of the semantic Web from a field of research and domain specific prototype applications to a widely used mainstream part of the Internet will be a step by step process, with researchers and developers building upon and adding to the body of knowledge and the work contributed by others. Many challenges remain for semantic Web developers such as efficient ontology integration and mapping, ontology versioning, natural language query processing, and the scalability of systems. The work undertaken here does not attempt to solve all of these challenges. Instead, it is hoped that a valuable contribution can be made to the field by which others can build upon. The paper presented an architecture for automated annotation and multiagent querying of semantic Web resources. The aim was to implement a practical approach to overcoming two of the major challenges of the semantic Web initiative which are i) availability of semantic Web content, and ii) the challenge of ontology based information retrieval. An automated Web site annotator was deployed to mark up Web sites with metadata consistent with a domain ontology. A system of multiagents was used to perform query functions. The Architecture was demonstrated in the accommodation services domain of the Australian Tourism Industry. Ultimately with further development, the system should be capable of handling user requests such as, “Arrange a one-week holiday for me, somewhere near the Great Barrier Reef Queensland (Australia), during September. Such a system would require ‘Car Hire’ and ‘Airline Booking’ agents to be developed. The coordination agent would also have to be modified accordingly. This will be the focus for the next phase of the research.

**References**


Copyright
Brooke Abrahams and Wai Dai © 2005. The authors assign to Victoria University (VU) and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to VU to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.