Semantic Web Services Methodology and Tool Extensions

Mohamed. S. Hajji* and Alaa. Al Maghrabi**

*Faculty of Computing and IT, University of Sohar, Oman.

**Independent Researcher, Netherlands.

Abstract

Web engineering developed strongly in the last few years. Some methodologies and graphical tools are well established now. However, according to survey conducted in this work, no systematic support for semantic web services SWS is provided.

In this work we developed special notations and diagrams as extensions to WebML to support SWS (this includes for example Intelligent Agents, Web Ontologies, UDDI registry, DAML-s matching engine, Ontologies database … etc.). Also a software tool to support the new extension and to automate the implementation was created. The tool provides the means to describe SWS in the standard way, besides the addition of semantic requests, search, semantic matching, and semantic result ranking. During the modeling process, we added notations specific to the composition and modeling of SWS.

Other known web engineering methods, like UWE and HERA, methodologies are considered in this research and their support, or lack of it, is discussed.

Keywords: Semantic Web Services, Modeling, Methodologies, WebML

INTRODUCTION

Software Engineering broadened its dimensions as web applications grew to cover a wide variety of sophisticated areas. Several methodologies like UWE [1] [2] and WebML [3] [4] are available with a main concern of supporting web application development.

Likewise, semantic web and semantic web services concepts evolved. Semantic web services enable the automation of Composition Publication, Discovery, and Selection of these services.

Different methodologies offered support in various ways to the representation of semantic web services and services’ components. Handling the semantic dimension implies the handling of various frameworks such as WSMF and DAML-S and new components like ontology and related representation languages. This implies consequently that the methodologies of web engineering should model these components in a comprehensible way, and at the same time support the automatic generation of the files of semantic web services description according to some frameworks.

Semantic web services are introduced in the second section of this paper. In addition, the section compares different web engineering methodologies. Related works are presented in section 3. Section 4 presents the main idea of this research discussing how WebML methodology was extended to incorporate semantic web services modeling. That implied the addition of new necessary notations and explanation of modeling steps with examples. Section 4 also reviews the features and properties of a software tool that we developed to generate semantic web services. The tool matches advertised services with the request and semantically ranks the results following a specific algorithm. Section 5 is dedicated for the conclusions and Future prospects.

BACKGROUND

Semantic web services:

The idea of web services blossomed out of the need to facilitate interaction and cooperation between web applications. Such cooperation implies search for services, linking, and publication which, traditionally, are done manually.

A directory called UDDI (Universal Description Discovery and Integration) is used for service publication. Programmers can search for services through the UDDI. When a programmer chooses a particular service assuming that it achieves his/her aim, he/she carries out manually the consuming process and the linking of the service with the desired application. WSDL (Web Service Definition Language) which is transferred using the protocol SOAP (Simple Object Access Protocol) are used in service-based Web application development. The framework of web services is presented in Figure (1) [5].

![Figure 1. Web Service Framework](image-url)
Augmented by the semantic dimension, web services gained the capability of automatic publication, discovery, selection and composition.

The semantic dimension is supported by Web Ontology Language (OWL-S), which is made of three ontologies. OWL-S is utilized to describe the tasks of semantic web services benefiting from WSDL descriptions and from UDDI for discovery [6]. Some researchers offered DAML-S (DARPA Agent Mark-up Language for Services) [7] as a framework for describing, explaining and constructing web services; to achieve that, this framework depends on DAML-S/UDDI Matchmaker. DAML-S employs specific matching algorithms in order to discover and construct semantic web services. In this respect the unification of ontologies among web services is necessary. The process of discovery starts by sending a message from the requiring service, and then a group of messages are initiated to determine the matching service (of the advertised ones). This is achieved through ontology analysis. Then the construction process is performed through ontologies unification. The ontologies of the two services are then combined. Semantic web services composition is not, however, within the scope of this work.

Methodologies:

Most web engineering methodologies focus on specific stages in web application development. Some methodologies concentrate on stages related to Hypertext Model and built around entity relationship modeling, like RMM (Relationship Management Methodology) [8] [9]. Other methodologies are more interested in the modeling of the adaptive web [10], and the addition of semantic concepts like Hera [11]. OOHDM (Object-Oriented Hypermedia Design Methodology) agrees with RMM about separating between stages (presentation-navigation). It is used to build enormous hypermedia web applications and it preserves the object-oriented style used in ordinary software development modeling [12]. It doesn’t, however, handle intelligent agents or semantic web issues.

UML Based Web Engineering (UWE) [1] [2] (which is an object-oriented methodology) covers the whole development lifecycle and gives good consideration for web application properties. It supports the automatic and semi-automatic generation of ready to use applications.

Web Modeling Language (WebML) [3] [4] takes the approach of modeling the components of web applications in a way very close to widespread development practices and uses notations appropriate for that purpose. WebML also supports automatic generation of web application and moved recently into supporting the development of mobile apps.

The structural approach of WebML consists of a group of models that can be integrated to support a high quality design.

We summarize the comparison between different methodologies properties including their support for different stages of development lifecycle in Table 1.

![Table 1. Comparison of web engineering methodologies](image)

### RELATED WORK

An early attempt to extend WebML to support web services can be found in [13]. Modeling of web services needs mainly the demonstration of messages between the consumer and service provider in addition to the modeling of the data recovered through the web service. This data can be stored temporarily or permanently in the application.

An example of how to compose an email message and recalls the remote web service in order to send that message is used to illustrate the idea. The process of sending shows the user starts from the page "send mail" and then enters the data which is then sent through the parameters of HTTP. HTTP here is considered as an input for "compose message" which transfers it into an XML file that plays the role of an input for "send mail". Eventually, the message is enveloped in a SOAP Message and sent to the remote web service. The message will be sent to an "error page" in the case of an error. The modeling of the output message analysis is also supported. In general, applications require data received from web services to be stored in a local repository in order to be used when service demand is over.

A work done by the team involved in the development of WebML describes a comprehensive approach to support semantic web applications including semantic web services [14] [15]. The work is based on the groups’ business processes BPMN (Business Process Management Notation) and supports partial automatic generation of the desired application. They emphasize the reliance on existing model driven software engineering methodologies and the already available WebML models. Figure 2 shows the development process of semantic rich web applications.
The approach handles modeling of the interaction between services by adopting BPMN, while web services interfaces are designed using visual diagrams based on the WebML models including specific hypertext primitives for Web Service invocation and publishing. The underlying data model, however, is designed using augmented ERD, this include modeling the local ontology of the application and the capability to import existing ontologies whenever they exist; the resulting set of ontologies is exposed to the underlying WSMX (Web Service Execution Environment).

In our work, though it can be seen as extension to WebML, we take a different approach with notations and symbols developed specifically to support the design and implementation of semantic web services where more “separation of concerns” is emphasized.

**CONTRIBUTION OF OUR RESEARCH**

*Expanding WebML with necessary notations :*

In addition to known WebML symbols, our modeling process needs new notations and symbols. Table 2 presents the new notations developed as part of our work.

The expansion is necessary for the representation of semantic web services and related processes. It permits us to represent the components of the DAML-S framework too.

**Justifications:** According to the framework based on DAML-S language, the process of semantic web services modeling needs the representation of the framework components and the messages that are exchanged between these components. The representation of the framework components is essential for us to do a full generation of the framework in the future, in addition to generating the ontologies which define the concerned services automatically.

<table>
<thead>
<tr>
<th>Table 2. New Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligent Agent</td>
</tr>
<tr>
<td>DAML-s matching engine</td>
</tr>
<tr>
<td>UDDI registry</td>
</tr>
<tr>
<td>Web Ontologies</td>
</tr>
<tr>
<td>SWS to Communication Model</td>
</tr>
</tbody>
</table>

**Figure 2.** The development process of semantic rich web applications [14]
Depending on DAML-S based platform, the mechanism of DAML-S/UDDI Matchmaker, and the description of semantic web services, we can claim that the notations in Table 2 are sufficient to represent semantic web services and messages between the components of web services. The modeling of all the stages of connection, search, composition and service advertising can be achieved using these notations. Modeling of detailed stages such as description transformation from and into the XML form and others is possible as well. Moreover through these notations, the representation of processes on ontologies like combining two ontologies or finding the shared concepts between them can be accomplished. These notations are adequate to develop diagrams that enable our tool to generate the framework DAML-S completely and to generate semantic web services ontologies in a standard way. These are the ontologies that we depend on to perform the matching between advertised and requested web services. The idea is clarified through an example in the following paragraphs.

It is not possible at this stage to be certain that the notations we developed are not over-detailed. Thorough usage and assessments with real life projects might be required to find out.

Semantic web services modeling in a DAML-S based framework:

We intend here to adopt the same pattern as in normal web services modeling after it is generalized to accommodate the new components of semantic web services and the new framework. Therefore, modeling of semantic web services within a framework that is based on DAML-S needs the following:

A- Message modeling.

B- The representation of retrieved data using SWS.

A- Messages modeling: at the start, we model the basic messages between the components of the framework DAML-S and services. These basic models can be seen as building blocks for any general system modeling.

The modeling of messages with description of symbols and mechanism can be explained as follows:

(1) Service registration in UDDI Registry: The user inputs the descriptions of a semantic web service (Figure 3); then the descriptions are handed to a specific adapter which transforms the descriptions into XML form. Ontologies which define the mentioned web service are, then, exported to the communication model whose main task is receiving service requests (service registration request, specific service request). The ontology is, then, handed to a UDDI converter which stores it in a UDDI registry after transforming it into the appropriate form.

(2) Requesting and searching for a service: The user, as shown in Figure 4, inputs the desired semantic web service descriptions which are handed to a special adapter that transforms the descriptions into XML. After that, the ontology which defines the target web service is sent to the communication model. The matching engine searches for the best service with the help of Web Ontology DAML. As a result, the most suitable service for the request is determined, and the request is sent to it immediately.

(3) Reply to request and combine the request and target services: Figure 5 shows target web service replying to the request, sending the reply message to the requesting service in XML form, and sending a signal to the communication model to combine the ontologies which define the two services and store them in UDDI Registry.
B- Modeling of data retrieved through web services: There is no difference in nature between the data retrieved through semantic web services and the data retrieved through ordinary web services; therefore, the process of modeling is the same. We illustrate by the following example an application that utilizes semantic web services needed to provide processes of transportation for consumers. In this example, which is widely used in web services publications, the user requests the service of transportation from the first available application which, in turn if it doesn’t have the required services, tries to use semantic web services of other applications. After exchanging the appropriate messages between the web service in the first application and services in other applications, the first application provides the user with offers as if they were its own. The processes of search and combination were, as expected from semantic web services, handled automatically.

Figure 6 shows the modeling of the sending process, while figure 7 presents the modeling of the receiving process.
The Tool:

Figure 8 illustrates our software tool that supports our extension to WebML methodology and the development of the diagrams of semantic web services.

The following capabilities and properties characterize this tool:

1. Diagram editor: We used the tool to develop the diagrams of the example mentioned above (Figures 3 to 7).
2. The tool can read and analyze owl-ontologies and store them in a specially tailored data structures, with support to navigation inside them.
3. It supports the generation of the whole framework DAML-S and its components, consequently allowing the matching between requested services and advertised ones.
4. Description of new semantic web services, their automatic generation, and then storing them in UDDI-Registry are fully supported with compliance with known standards.
5. The tool enables the addition of semantic requests and the search for semantic web services.
6. Furthermore, the tool enables semantic matching between the request and already stored services, and provides ranking of the search results.

The automatic generation supported by the tool is only for semantic web services, and not the full semantic web application.
CONCLUSIONS AND FUTURE WORK

Several current research works are focusing on to semantic web services related topics. In this research, we developed an extension to WebML methodology in order to enable the modeling and design of semantic web services. We augmented WebML with special notations, symbols and diagrams. In addition, a software tool was developed to support the new extension and to automate the implementation. The tool provides support for the description of semantic web services in the standard way, the addition of semantic requests, search, semantic matching, and semantic result ranking.

We added notations specific to the processes of modeling and composition of semantic web services, and left to future research the process of achieving and finishing the composition.

REFERENCES


