

# Radiant: A tool for semantic annotation of Web Services

Karthik Gomadam, Kunal Verma, Douglas Brewer, Amit P.Sheth, John A. Miller

Large Scale Distributed Information Systems Lab,  
Department of Computer Science, University Of Georgia, Athens.  
{karthik, verma, brewer, amit, jam}@cs.uga.edu

## Abstract

Web processes are next generation workflows created using Web services. Semantic representation of Web services is essential to achieve dynamic composition and configuration of Web processes. We demonstrate the Radiant system for annotating and creating semantic Web services using WSDL-S.

**Key Words:** Semantic Web Services, Semantic Web Service Annotation, Semantic Web Service Publication, Semantic Web Service Discovery, WSDL-S, METEOR-S

## 1. Introduction:

Service Oriented Architecture based on XML messaging and loose coupling, has the promise to realize more agile and dynamic Web processes. However, the current standards lack sufficient semantics to realize dynamism despite recent business use cases [1] illustrating the need for such dynamic capabilities. The key to semantic Web services is having semantics in the service description itself. One approach is to develop semantic Web services, where services are annotated based on ontologies. These capabilities can improve service discovery and facilitate for more dynamism in process composition and configuration. An approach to achieve dynamic process configuration using semantic Web services is discussed in [2]. Significant research has been done in the area of semantic Web services, resulting in frameworks like OWL-S and WSMO, and specifications like WSDL-S.

The rest of the paper is organized as follows. Section 2 discusses the WSDL-S specification for adding semantics to Web services. Section 3 discusses the Radiant tool .

## 2. WSDL-S

In this section, we provide a brief introduction to WSDL-S, a joint IBM-UGA effort [3]. This approach is based on using extensibility elements to add semantics to the existing standard as described in Table 1.

Extension Element/Attribute	Description
<i>modelReference</i> (Parent Element: Message, Complex Type, Element)	Allows for one-to-one associations of WSDL input and output type schema elements to the concepts in a semantic model.
<i>schemaMapping</i> (Parent Element: Complex Type, Element)	Allows for many-to-many associations of WSDL input and output type schema elements to the concepts in a semantic model – typically associated with XML schema complex types.
<i>Action</i> (Parent Element: Operation)	Captures the semantics of the functional capabilities of an operation.
<i>pre-conditions</i> (Parent Element: Operation)	Set of semantic statements (or expressions represented using the concepts in a semantic model) that are required to be true before an operation can be successfully invoked
<i>Effects</i> (Parent Element: Operation)	An effect is a set of semantic statements (or expressions represented using the concepts in a semantic model) that must be true after an operation completes execution.
<i>servicecategorization</i> . (Parent Element: Operation)	Service categorization information that could be used when publishing a service in a Web Services registry such as UDDI.

Table 1: WSDL-S extensions

WSDL-S takes a purely evolutionary approach that extends the current standards. This makes it possible for existing services to be semantically annotated without much of an overhead. The framework augments the expressivity of Web service descriptions using ontologies. However the framework is agnostic to ontological representation language. This makes it possible for annotating services with different ontologies, written in different representation languages. Due to the relevance to the key challenge of semantic mappings which are key to integration and composition and the WSDL-S's ontology based interoperability support, it has seen significant interest among researchers and industry alike.

### 3. Radiant

The Radiant semantic Web service development plug-in for Eclipse toolkit allows users to annotate WSDL files with the OWL ontologies. The extensibility attributes defined in WSDL-S, which are shown in Table 1, are used to add these annotations. Annotation of existing WSDL documents with concepts from one or more ontologies is supported in this tool based partly on the early work described in [5]. A snapshot of the plug-in is shown in Figure 1.

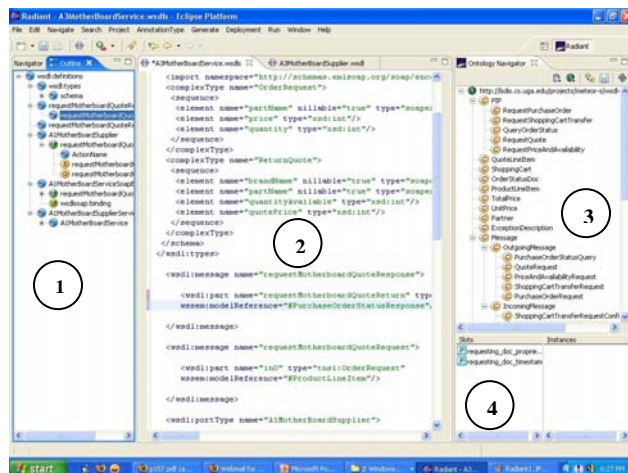


Figure 1: Radiant plug-in for WSDL-S

The tool offers tree representations of the source WSDL files (1) and OWL files (3) and (4) simultaneously to enable the user to choose the most appropriate mapping. The tool offers drag and drop functionality. This allows users to drag concepts from the ontology and drop it at the desired location either in the WSDL tree or in the WSDL-S file. The WSDL-S file is depicted in (2). The properties of ontology classes are shown in (4). The plug-in also allows the user to load multiple ontologies. Moreover, the tree representations help the user to browse/navigate through the entire document in a comparatively less amount of time, without having to deal

with Java code or XML syntax. The tool essentially frees the developer from the task of representing and incorporating annotations and helps him/her focus on the task at hand, to provide the most appropriate annotations. The tool also shows the annotated file in the text editor.

### 4. Conclusion

In this work we have presented a tool for annotating Web services with ontologies. Radiant plug-in allows the users to add the WSDL-S extensibility attributes to existing Web services. This allows for creating semantic Web services with less overhead. We believe that by creating tooling support to the WSDL-S specification the adoption of semantic Web services can be made easier. Potential future work includes integrating the Radiant plug-in with the METEOR-S Web Service Annotation Framework (MWSAF) [5], for enabling semi-automatic annotation.

We would like to thank Rama Akkiraju and Joel Farrel of IBM and Meenakshi Nagarajan of LSDIS Lab, UGA for their work in creating the joint UGA-IBM WSDL-S specification.

### References

1. Integrated Shipping Co Integrated Shipping Environment Consortium Use Case, available at [http://www-306.ibm.com/software/ebusiness/jstart/casestudies/niip\\_isec.s.html](http://www-306.ibm.com/software/ebusiness/jstart/casestudies/niip_isec.s.html)
2. Kunal Verma, Karthik Gomadam, Amit P. Sheth, John A. Miller, Zixin Wu, The METEOR-S Approach for Configuring and Executing Dynamic Web Processes, LSDIS Technical Report, 2005.
3. R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M. Schmidt, A. Sheth, K. Verma, Web Service Semantics - WSDL-S, A joint UGA-IBM Technical Note, <http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.html>
4. RosettaNet ontology for Supply Chain Domain. <http://lsdis.cs.uga.edu/projects/meteor-s/wsdls/ontologies/rosetta.owl>
5. Abhijit Patil, Swapna Oundhakar, Amit Sheth, Kunal Verma, METEOR-S Web service Annotation Framework, The Proceedings of the Thirteenth International World Wide Web Conference, May, 2004 (WWW2004), pp. 553-562

## 5. Demonstration Overview

A live software demonstration of the system will be presented. This will include creating WSDL-S using Radiant plug-in as well as the METEOR-S semantic Web service publication and discovery interface.

The Radiant plug-in demonstration will show the capabilities of the system to annotate WSDL documents. For introductory demonstration, we will use the RossettaNet ontology [4] and WSDL documents from the domain of supply chain. A detailed snapshot of the Radiant plug-in is illustrated in Figure 2. The WSDL tree view is shown in (1). The WSDL-S document is illustrated in (2). OWL ontology tree view is presented in (3). The OWL properties are displayed in (4).

The annotated services will then be published in semantically enhanced UDDI using the discovery and publication interface. The plug-in can be downloaded for use from

<http://lsdis.cs.uga.edu/projects/meteor-s/downloads/>.

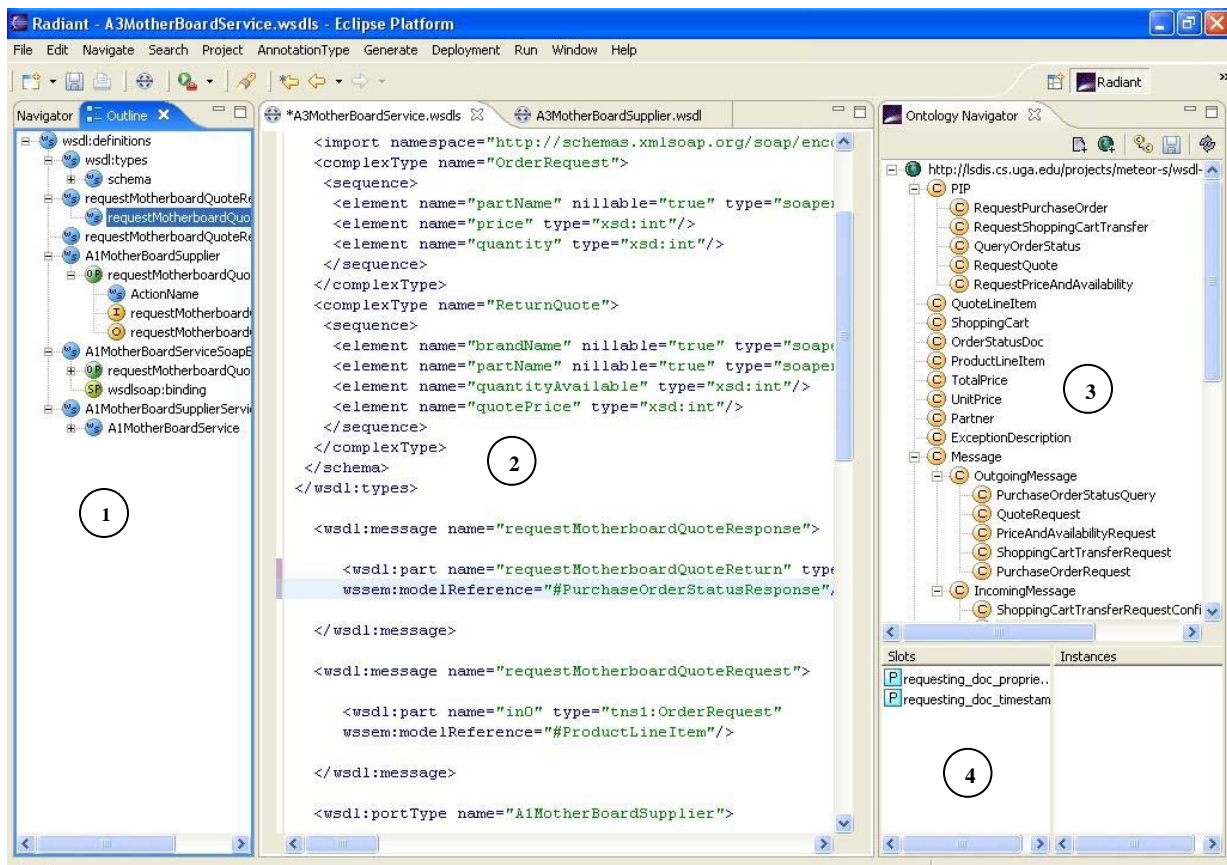


Figure 2: Detailed Snapshot of Radiant plug-in for WSDL-S