A Plug-in for Adding Semantics to WSDL

Preeda Rajasekaran, Kunal Verma, John Miller and Amit Sheth

LSDIS Lab, Computer Science Department, University of Georgia, Athens, 30602
{preeda, verma, jam, amit}@cs.uga.edu

Abstract

This paper describes an Eclipse plug-in developed as a part of METEOR-S for supporting Semantic Web Services. A brief introduction on the background of Semantic Web Services is given to highlight the importance of the plug-in. The features of the plug-in and its role in Semantic Web Applications development are also presented. WSDL-S, which extends WSDL with semantic annotations, has been provided as an input to WSDL committee, the tool has been demonstrated to our collaborators at IBM, an alpha version has been released with test cases, and a refined version with support for semi-automatic semantic annotation will be released in near future.

1 Introduction

The advent of web services has revolutionized many aspects of e-business. The potential of Web Services to offer dynamic integration between disparate systems has proved as a valuable asset in Business-to-Business (B2B) e-commerce applications. The current model of Web services is based on XML based syntactical standards like WSDL and UDDI which typically require human intervention in the various stages of Web services based system development.

Recently, the Semantic Web has been proposed as an approach to significantly automate the usage of Web-based resources, which includes static content like data and documents as well as dynamic content like applications represented as Web services. Representing semantics using formal languages such as W3C's OWL [1] can help express the meaning of a resource, in a machine understandable manner. Machine understandability of a Web resource not only helps in automating a number of tasks, but also serves to improve the efficiency of searching, filtering and categorizing information on the web. Semantic Web Services are Web Services whose descriptions are enhanced with semantic annotations. In order to enhance WSDL with semantic annotations, we have proposed a language called WSDL-S [2] which incorporates semantics with the help of extensibility features of WSDL 2.0 along with some minor changes. The plug-in 'METEOR-S WSDL-S Generator' discussed in this paper facilitates the creation of WSDL-S documents from WSDL 1,1 documents by allowing users to manually annotate the WSDL elements with ontology concepts.

2 Web Service and Semantic Web Services

Web Services can be defined as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web'. The main components of Web Services are a description language WSDL [3], a messaging protocol SOAP [4] and standard of discovery and publication UDDI [5]. The current discovery model is based on finding appropriate Web services using the search mechanism provided by UDDI. This limits discovery to either having prior knowledge of UDDI technical models (requires human involvement) or a keyword based search (characterized by low precision and high recall). A greater level of automation and better

search is possible with the help of semantic annotations.

Semantic Web Services are Web Services whose descriptions are enhanced with semantic annotations to provide unambiguous representation of Web service functionality and its parameters using a formal representation language. METEOR-S from the LSDIS lab from the University of Georgia, is one of the leading research initiatives, along with OWL-S [6] and WSMO [7], for in realizing semantic Web services.

In an environment where business partners change frequently, application integration between components can be done smoothly if the semantics of the operation and associated concepts are available.

3 METEOR-S

The METEOR (Managing End-To-End OpeRations) project at the LSDIS Lab, University of Georgia, focused on workflow management techniques for transactional workflows. Its follow on project, which incorporates workflow management for Semantic Web Services is called METEOR-S [8] (METEOR for Semantic Web Services). A key feature in this project is the usage of semantics for the complete lifecycle of semantic Web processes, which represent complex interactions between semantic Web services.

The main stages of creating semantic Web processes have been identified as development, annotation [9], discovery, composition and orchestration [10]. A key research direction of METEOR-S has been exploring different kinds of semantics, which are present in these stages. We have identified data, functional, Quality of Service and execution semantics as different kinds of semantics and are working on formalizing their definitions. A detailed explanation of the underlying conceptual foundation of METEOR-S is present in [11, 12, 13, 14].

4 WSDL-S

METEOR-S takes in semantic annotations via three types of files [13], one of which is WSDL-S. The WSDL-S document is a semantically enriched (annotated) WSDL 2.0. .

One of the central purposes of WSDL is to describe interfaces (formerly known as port-types) for Web services. In general, service providers/implementers could use a standard interface, extend a standard interface or develop their own. Broadly speaking, an interface contains a set of operations. Each operation has a signature, which includes an operation name, input, output and exception messages. These messages have types that are defined using XML schema in WSDL 1.1. In an effort to make Web services more flexible, WSDL 2.0 allows using any type system. We use this feature of WSDL 2.0 to create WSDL-S which allows the use of OWL as a type system.

Complete details of WSDL-S as well as its meta-model is present in [1]. In this section, we will briefly describe the changes which are used to enhance WSDL 2.0 with semantics.

1. "**Action**" attribute is added to operation tag. It depicts the action the operation performs with the help of ontology concepts..
2. "**Pre**" and "**Post**" tags are added as children of the operation tag. They are used to depict pre and post conditions with the help of a rule language like SWRL [15].
3. The use of OWL and XMI types is proposed for the values of the "**Element**" attribute
4. "**Exception**" tag is added as a child of the operation tag. It is used to represent the exceptions that can be thrown by the operation with the help of ontological concepts.

A sample WSDL-S file is shown in Appendix A.

5 Plug-In: METEOR-S WSDL-S Generator

The plug-in 'METEOR-S WSDL-S Generator' helps to generate WSDL-S files from WSDL file via manual annotation of semantic concepts. As discussed in the previous section, to enable semantic web services we need to include semantics in the description of the services.. WSDL files provide service descriptions at the syntactic level.

With the help of extensible tags the semantic information can be incorporated into WSDL files. METEOR-S adapts OWL to serve as the source of semantic concepts. WSDL files are annotated using concepts from OWL ontologies.

5.1 Functionality and Features

Appendix B shows the interface of the plug-in. The plug-in features are incorporated into Eclipse workbench tool-bar and menu-bar.

1. The plug-in helps to open-up WSDL files in the form of a tree (as shown in callout 1), for easy perusal of WSDL documents. The operations in a WSDL file are grouped by the port-type they are associated with. Inputs and output of each operation are displayed along with their corresponding data-type information..

2. The semantic information is provided via OWL files. The OWL ontology is represented as a tree on the right-hand-side (as shown in callout 2). The tree representation of the ontology, displays the properties and descriptions associated with the concept. Double-clicking the concept helps to bring up the details associated with a concept.

3. To annotate a WSDL Element it has to be chosen from the WSDL tree (as shown in callout 3) and the corresponding ontological concept is chosen from the ontological tree (as shown in callout 4).The mapping between the concepts is established them using 'Save Mappings'.

This tool aids users, who want to take advantage of the features offered by Semantic Web Services, by offering assistance to annotate WSDL files with semantic descriptions. While this task can be done manually by editing the WSDL files, using the plug-in reduces the possibilities of errors while editing complex WSDL files. Moreover the tree representation of the WSDL files and OWL files helps the user to browse/navigate through the entire document in a comparatively less amount of time, without having to deal with XML syntax representations. The plug-in further reduces errors by validating WSDL files prior to annotation.

Warnings are issued if the user decides to generate a partially annotated file, as automated discovery and composition will be successful only if complete semantic information of the service/operation is available.

6 Conclusion and Future Work

Ongoing work on this plug-in involves integrating it with MWSAF [9] to perform semi-automated annotation of WSDL files. A more expressive graphical interface for ontology and WSDL file display will be explored, which will enable the user in choosing the most appropriate concepts for annotation. The plug-in will be improved to provide features extending support for the generation of a variety of semantic descriptions of web service e.g. annotated WSDL1.1 code, annotated Java Source Code [13]. The WSDL-S specification itself has been provided to the WSDL standardization committee for consideration.

This work was partly funded by IBM's Eclipse Innovation Grant Program and this plug-in was demonstrated to the Semantic e-Business Middleware Group at IBM T.J. Watson Research Center, Hawthorne, NY. First version of this plug-in is available for download at http://lsdis.cs.uga.edu/METEOR-S/ WSDL-S-Generator, and a revised version will be shortly posted.

References

[1] McGuinness et al., Web Ontology Language (OWL), Web-Ontology (WebOnt) Working Group, http://www.w3.org/2001/sw/WebOnt/, 2002

[2] WSDL-S :A Proposal to W3C WSDL 2.0 Committee, http://lsdis.cs.uga.edu/projects/WSDL-S/wsdl-s.pdf

[3] Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[4] A. SOAP Version 1.2 Part 0: Primer, W3C Recommendation 24 June 2003, http://www.w3.org/TR/2003/REC-soap12-part0-20030624/

[5] UDDI Version 2 Specifications http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv2

[6] Martin, D.L., McDermott, D., McIlraith, S.A., Narayanan, S.,Paolucci, M., Payne T.R., and Sycara, K. The DAML Services Coalition, "DAML-S: Web Service Description for the Semantic Web", *The First International Semantic Web Conference ,(ISWC 2002)*, Chia, Sardegna, Italy, 2002

[7] D.Roman et. al., WSMO –Web Service Modeling Ontology (WSMO),DERI Working Draft14Feb2004,http://www.wsmo.org/2004/d2/v0.1/20040214/

[8] METEOR-S :Semantic Web Services and Processes, http://swp.semanticweb.org, 2002-2004

[9] A. Patil, S. Oundhakar, A. Sheth, K. Verma, METEOR-S Web service Annotation Framework, *Proceeding of the World Wide Web Conference*, pages 552-562, New York, NY, May 2004.

[10] K. Verma, R. Akkiraju, R. Goodwin, P. Doshi, J. Lee, On Accommodating Inter Service Dependencies in Web Process Flow Composition, *AAAI Spring Symposium on Semantic Web Services*, pages 37-43, 2004

[11] A. Sheth, "Semantic Web Process Lifecycle: Role of Semantics in Annotation, Discovery, Composition and Orchestration," Invited Talk, *WWW 2003 Workshop on E-Services and the Semantic Web*, Budapest, Hungary, May 20, 2003.

[12] K. Sivashanmugam, K., Verma, A. Sheth, J. Miller, Adding Semantics to Web Services Standards, *Proceedings of 1st International Conference of Web Services*, pages 395-401, LasVegas, Neveda, 2003.

[13] P. Rajasekaran, J.Miller, K.Verma, A.Sheth, Enhancing Web Services Description and Discovery to Facilitate Orchestration, *In the Proceedings of SWSWPC '04*, pages 34-37, San Diego, California, 2004

[14] R.Aggarwal, K.Verma, J.Miller, W.Milnor, "Constraint Driven Web Service Composition in METEOR-S," *Proceedings of the 2004 IEEE International Conference on Services Computing (SCC 2004),* Shanghai, China (September 2004).

[15] Horrocks et al., SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.daml.org/rules/proposal/, 2004

Appendix A: Generated WSDL-S File

(Semantic annotations in WSDL-S are shown in bold.)

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
        name = "BatterySupplier"
        targetNamespace = "http://lsdis.cs.uga.edu/meteor/BatterySupplier.wsdl20"
        xmlns = "http://www.w3.org/2004/03/wsdl"
        xmlns:tns = "http://lsdis.cs.uga.edu/BatterySupplier.wsdl20"
        xmlns:rosetta = " http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/pips.owl "
        xmlns:mep=http://www.w3. rosetta:PurchaseOrderStatusResponse org/TR/wsdl20-patterns>
  <interface name = "BatterySupplierInterface"
         description = "Computer PowerSupply Battery Buy Quote Order Status "
         domain="naics:Computer and Electronic Product Manufacturing" >
        <operation name = "getQuote" pattern = "mep:in-out" action = "rosetta:#RequestQuote" >
          <input messageLabel = "qRequest" element = "rosetta:#QuoteRequest" />
          <output messageLabel = "quote" element = "rosetta:#QuoteConfirmation" />
```

```
  </operation>
  <operation name = "placeOrder" pattern = "mep:in-out" action = "rosetta:#RequestPurchaseOrder" >
     <input messageLabel = "order" element = "rosetta:#PurchaseOrderRequest" />
    <output messageLabel = "orderConfirmation" element = "rosetta:#PurchaseOrderConfirmation" />
    <exception element = "rosetta:#DiscountinuedItemException" />
     <pre condition = " order.PurchaseOrder.PurchaseOrderLineItem.RequestedQuantity > 7" />
  </operation>
  <operation name = "checkStatus" pattern="mep:in-out" action = "rosetta:#QueryOrderStatus" >
     <input messageLabel = "statusQuery" element = "rosetta:#PurchaseOrderStatusQuery" />
      <output messageLabel = "status" element = "rosetta:#PurchaseOrderStatusResponse" />
      <exception element = "rosetta:#OrderNumberInvalidException" />
   </operation>
 </interface>
</definitions>
```

Appendix B: Plug-in UI