

## Web Service Semantics - WSDL-S

R. Akkiraju<sup>2</sup>, J. Farell<sup>2</sup>, J.A. Miller<sup>1</sup>, M. Nagarajan<sup>1</sup>, A. Sheth<sup>1</sup> and K. Verma<sup>1</sup>

<sup>1</sup>LSDIS Lab, University of Georgia and <sup>2</sup>IBM

### Abstract

Enterprise application integration and business process management are critical to most organizations today. In the context of Service Oriented Architecture, this translates into the technical challenges of discovery, reuse and composition of services. In implementing service oriented architectures, Web Services are becoming an important technological component. However, Web Service Description Language [WSDL] standard does not contain the semantic expressivity needed to represent the requirements and capabilities of Web Services – a requirement for addressing the vexing heterogeneity challenges that need to be addressed for achieving (semi) automated discovery, improved reuse and faster composition. While efforts by the Semantic Web community to address this issue have resulted in proposals such as WSMF [FB02], OWL-S [OWL-S] and WSMO [WSMO], they are seen as revolutionary and do not build on current WSDL standards, thereby making industry adoption less likely, since industry prefers evolutionary approaches whenever available. We believe and demonstrate that any approach to annotating Web Services should and could be built by enriching the current Web Services standards with semantic descriptions. In particular, we summarize an approach to annotating Web Services [AFM+05] which builds on current standards while allowing for the use of multiple semantic representation languages to annotate services.

### Introduction

Integration is the most critical issue facing many companies today. Companies are forced to be flexible and adaptive to the changes in the market conditions while keeping their IT expenses in check. The problems with integration are multi-faceted involving people, data, and processes. In the context of a large enterprise, typically systems are developed over different periods of time, for different initial purposes, by different organizations or units within an enterprise and with different structures and vocabulary. This leads to substantial heterogeneity in syntax, structure and semantics, even if the Service Oriented Architecture is used. For example, what one service interface in one system may encode as *itemID*, *dueDate*, and *quantity* may be referred to by another service interface in a different system as *UPC* (Universal Part Code), *itemDeliveryTime* and *numberOfItems*. At the heart of data and process integration is the need to resolve the similarities and differences among various formats, structures, interfaces and ultimately vocabulary. Developing automated tools to help resolve these types of syntactic, structural and semantic similarities and differences is key to keeping IT expenses in check. In developing automated tools that can perform semantic and syntactic similarity matching, we need to address two fundamental issues. First, we need a way to describe the capabilities and requirements of services. Second, we need effective and efficient matching systems to match the requirements and capabilities of services i.e, identify similarity between services. In this paper we address the former of the two issues mentioned, which will then enable a solution to the latter issue.

Recent industry activity in Web Services standards [WSDL 2001] has reinvigorated the business process integration community. By providing a standards-based framework for exchanging information dynamically on demand between applications, Web Services show promise to address the process integration needs of enterprise application integration. However, WSDL, in its current form, suffers from the lack of semantics leaving the promise of automatic integration of applications written to Web services standards unfulfilled. Adding semantics to represent the requirements and capabilities of Web services is essential for achieving automation in service discovery and execution. In fact, the need for semantics is pervasive in the complete lifecycle of Web services, encompassing description, discovery, composition, and choreography/orchestration, and involving a broad variety of semantics covering data, function/behavior, quality of service and execution [WSML, S03, SVSM03].

During development, the service provider can explicate the intended semantics by annotating the appropriate parts of the Web service with concepts from a richer semantic model [POSV04, HJK04]. Since semantic models provide agreement on semantics of terms, and may provide formal and informal definitions of the entities, there will be less ambiguity in the intended semantics of the provider. During discovery, the service requestor can describe the service requirements using terms from the semantic model. Semantic matching and reasoning techniques can be used to find the semantic similarity between the service description and the request [SPAS03, SMSV05]. During composition, the functional aspect of the annotations can be used to aggregate the functionality of multiple services to create useful service compositions. Finally during invocation, mappings to the semantic model can be used to enable data transformations. Therefore, once represented, semantics can be leveraged by tools to automate service discovery, mediation, composition and monitoring.

One of the ways the semantic Web community is working to address these issues is by developing a semantic markup language for Web Services such as OWL-S [OWL-S] and WSML [WSML], based on description logic [DL] and F-logic [F-Logic] concepts, respectively. While the semantic expressivity is rich in these approaches, they require the creation of new semantic models of services on top of the syntactic WSDL specification of a service. Also, they assume that everyone uses OWL [OWL] for representing ontologies. While OWL is a W3C standard for ontology representation, there are other knowledge representation languages such as Topic Maps [TM] and Common Logic [CL]. Moreover, “organizations developing ontologies may prefer to build on legacy data and process models represented in UML” [ODM]. To support the use of potential multiple modeling languages, we propose a new and inclusive approach to add semantic annotations to WSDL documents in our WSDL-S technical note document [AFM+05], which significantly revises our earlier preliminary approach which originally advocated for an evolutionary approach [SVSM03].

## Requirements for Web Services Semantics

We recommend that certain principles guide any work to define a framework for Web services semantics. Ours is guided by the following.

- *Build on existing Web Services standards:* The Web services standards are fast becoming a preferred technology for application integration because of the promise of their interoperability. Companies are making investments in integration projects based on Web Services. Therefore, we believe that any approach to adding semantics to Web Services should be specified in an upwardly compatible manner so as to not disrupt the existing install-base of Web Services and associated investments in human training and technical solutions [Shivashanmugam].
- *The mechanism for annotating Web services with semantics should be independent of the semantic representation language:* There are a number of potential languages for representing semantics such as OWL [OWL], WSMO [WSMO], and UML [UML]. Each language offers different levels of semantic expressivity and developer support. Our position is that it is not necessary to tie the Web services standards to a particular semantic representation language. This is consistent with the approach prescribed by Sivashanmugam et al in their work [Sivashanmugam]. By keeping the semantic annotation mechanism separate from the representation of the semantic descriptions, the approach offers flexibility to developer community to select their favorite semantic representation language. In the next section, we will show a way such independence can be achieved.
- *The mechanism for annotating Web services with semantics should allow the association of multiple annotations written in different semantic representation languages:* As mentioned earlier, there are many potential semantic representation languages. Service providers may choose to annotate their services in multiple semantic representation languages to be discovered by multiple discovery engines. Therefore, we believe that the mechanism for annotating Web Services with semantics should allow multiple annotations to be associated with Web Services.
- *Support semantic annotation of Web Services whose data types are described in XML schema:* A common practice in Web services-based integration is to reuse interfaces that are described in XML. The definition of

business documents using XML schema is a wide-spread and successful practice. XML schemas will be an important data definition format for the foreseeable future. We believe that the semantic annotation of service inputs and outputs should support the annotation of XML schemas. WSDL 2.0 supports the use of other type systems in addition to XML Schema, so constructs in semantic models, such as classes in OWL [OWL] ontologies, could be used to define the Web service input and output data types. But an approach that does not address XML schema-based types will not be able exploit exiting assets or allow the gradual upgrade of deployed WSDL documents to include semantics.

- *Provide support for rich mapping mechanisms between Web Service schema types and ontologies:* Given our position on the importance of annotating XML schemas in Web service descriptions, attention should be given to the problem of how to map XML schema complex types to ontological concepts. Again, an agnostic approach to the selection of schema mapping languages is called for. For example, if the domain model is represented in OWL, the mapping between WSDL XSD elements and OWL concepts can be represented in any language of user's choice such as: XSLT, XQuery, RDF/S, OWL or any other arbitrary language as long as the chosen language is fully qualified with its own namespace.

## Web Services Semantics: WSDL-S Approach

Following the principles described above IBM and University of Georgia have collaboratively developed Web Services Semantics – WSDL-S [AFM+05] – a technical note document. This technical note prescribes a mechanism to annotate Web Services with semantics. It is conceptually based on, but a significant refinement in details of, the original WSDL-S proposal [MVR+04] from the LSDIS laboratory at the University of Georgia. In WSDL-S, we augment the expressivity of WSDL with semantics by employing concepts similar to those in OWL-S while being agnostic to the semantic representation language (we only refer to profile model in this document. OWL-S process model compares with BPEL4WS and it is not discussed here). The advantage of this evolutionary approach to adding semantics to WSDL is multi-fold. First, users can, in an upwardly compatible way, describe both the semantics and operation level details in WSDL- a language that the developer community is familiar with. Second, by externalizing the semantic domain models, we take a language-agnostic approach to ontology representation. This allows Web service developers to annotate their Web services with their choice of modeling language (such as OWL, or legacy models developed in UML or other knowledge representation languages discussed above). This is significant because the ability to reuse existing domain models expressed in modeling languages like UML can greatly alleviate the need to separately model semantics. Finally, it is relatively easy to update the existing tooling around WSDL specification to accommodate our incremental approach. While it is noted that the theoretical underpinnings of OWL-S in description logic or WSMO in F-Logic make them rich languages for representing semantics, we believe that extending the industry standards such as WSDL to include semantics is a more practical approach for adoption. Moreover, by externalizing the semantic domain models in our proposal, we still allow for richer representations of domain concepts and relationships in languages such as OWL, thereby bringing together the best of both worlds. Use of expressive mapping representation and techniques can further enable this approach to deal with significant types of syntactic, structural, representational and semantic heterogeneity.

Below, we illustrate the key elements of our approach to adding semantics to the WSDL specification and illustrate it with an example.

### Using the Extensibility Elements of WSDL

In this section we briefly describe how semantic annotations are added to WSDL document elements in the Web Services Semantics – WSDL-S technical note (for details, see the technical note [AFM+05]). In this approach, we focus on semantically annotating the abstract definition model of a service in WSDL specification to enable dynamic discovery services. Annotation of service implementation model is not discussed for brevity. In essence, we provide URI reference mechanisms via extensibility elements to the interface, operation and message constructs

to point to the semantic annotations defined in the externalized domain models for services. A quick summary of the extensibility elements provided in this technical note are:

- an extension element, namely *wssem:modelReference*, to allow for one-to-one associations of WSDL input and output type schema elements to the concepts in a semantic model
- an extension attribute, namely *wssem:schemaMapping*, to allow 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
- two new elements, namely *wssem:precondition* and *wssem:effect*, which are specified as child elements of the *operation* element and describe the semantics of the operation along the lines of the OWL-S approach. Preconditions and effects are primarily used in service discovery, and are not required to invoke a given service, and
- an extension attribute on interface element, namely *wssem:serviceCategorization*. It consists of service categorization information that could be used when publishing a service in a Web Services registry such as UDDI. It corresponds to the categorization concept proposed in OWL-S.

An example WSDL document that is semantically annotated using this approach is given below (again see the technical note for details). In this sample, we present a simple purchase order service. The inputs and outputs of ProcessPurchaseOrder service are annotated with semantic references and preconditions and effects are introduced as extensibility elements to the ProcessPurchaseOrder operation. The semantic concepts and their relationships are modeled in an OWL ontology – PurchaseOrder.owl (presented in Appendix C of [AFM+05]). Annotating XSD elements with OWL concepts represents a schema mapping problem with representation model heterogeneity, as OWL is far more expressive than XSD. A survey of schema matching and mapping approaches, as well as description of some approaches which have tried to automate the XSD to OWL approach is present in Appendix D (AFM+05). A revised version of the LSDIS semantic annotation tool that supports this proposal is expected in May 2005.

```
<definitions name="PurchaseOrder"
  .....
  xmlns:wssem="http://www.ibm.com/xmlns/WebServices/WSSemantics"
  xmlns:POOntology="http://www.ibm.com/ontologies/PurchaseOrder.owl">

  <types>
  .....
  <xs:element name=" processPurchaseOrderRequest" type="tProcessPurchaseOrderRequest"/>
  <xs:complexType name="tprocessPurchaseOrderRequest">
    <xs:sequence>
      <!--Semantic annotations for these complex types are given in their respective type definitions -->
      <xs:element name="billingInfo" type="xsd1:POBilling"/>
      <xs:element name="orderItem" type="xsd1:POItem"/>
    </xs:sequence>
  </xs:complexType>
  <!--Semantic annotation is added directly to leaf element in a complex type -->
  <xs:element name=" processPurchaseOrderResponse" type="xs:string"
    wssem:modelReference="POOntology#OrderConfirmation"/>
  </xs:schema>
</types>

<interface name="PurchaseOrder">
  <operation name="processPurchaseOrder" pattern=http://www.w3.org/2004/03/wsd/in-out >

  <input messageLabel = "processPurchaseOrderRequest" element="tns:processPurchaseOrderRequest"/>
  <output messageLabel = "processPurchaseOrderResponse" element="processPurchaseOrderResponse"/>

  <!--Precondition and effect are added as extensible elements on an operation -->
  <wssem:precondition name="PreExistingAcctPrecond" modelReference="POOntology#AccountExists">
  <wssem:effect name="ItemReservedEffect" modelReference="POOntology#ItemReserved"/>
```

```

</operation>
</interface>
</definitions>

```

Semantic annotation for POItem.xsd is show below.

```

<?xml version="1.0" encoding="UTF-8"?>
....
  xmlns:wssem="http://www.ibm.com/xmlns/WebServices/WSSemantics"
  xmlns:POOntology="http://www.ibm.com/ontologies/PurchaseOrder.owl">

<import location="WSSemantics.xsd" namespace="http://www.ibm.com/xmlns/WebServices/WSSemantics/" />

  <complexType name="POItem" >
    <all>
      <element name="itemDeliveryDate" nillable="true" type="dateTime"
        wssem:modelReference="POOntology#DueDate" />
      <element name="numItems" type="float"
        wssem:modelReference="#POOntology#Quantity" />
      <element name="ItemID#" nillable="true" type="string"
        wssem:modelReference="POOntology#UPC" />
      <element name="desc" nillable="true" type="string"
        wssem:modelReference="POOntology#ItemDesc" />

    </all>
  </complexType>
</schema>

```

## Conclusions

Web Services are becoming an important technological component in the application integration domain. Large enterprises are deploying as many as several thousand Web services, a situation that brings into focus the need for tools to discover and integrate these services with each other and with the other applications in use. Semantics play a crucial role in the development of these tools as they enable dynamic discovery, composition and execution of Web Services. Web Services standards, which currently lack semantics, should be extended to include semantics to enable this automation. However, careful consideration must be given to develop a framework for Web Services semantics that is practical. We believe that any approach for Web Services semantics should build on the current Web Services standards and allow for using the semantic representation language of the user's choice. Standardization in the area of Web Service semantics calls for a careful and inclusive approach, given the early stage of adoption of semantic representation languages, and the availability of multiple potential candidates. Also, the approach has to be based on experience and lessons learned in applying the approach to solve real-world integration problems.

## References

- [AFM+05] Akkiraju, R., Farrell, J., Miller, J.A., Nagarajan, M., Schmidt M-T., Sheth, A., Verma, K. Web Service Semantics---WSDL-S, Technical Note, Version 1.0, April 2005, <http://lsdis.cs.uga.edu/Projects/METEOR-S/WSDL-S> and <http://www.alphaWorks.ibm.com/xxx>
- [FB02] Fensel, D. and Bussler, C. The Web Service Modeling Framework WSMF. In: Electronic Commerce Research and Applications, Vol. 1, Issue 2, Elsevier Science B.V., Summer 2002.
- [KLW95] Kifer, M., Lausen, G., Wu, J. Logical Foundations of Object Oriented and Frame Based Languages, Journal of the ACM, Vol 42, 1995, p. 741-843, <http://www.informatik.uni-freiburg.de/~dbis/Publications/95/flogic-jacm.html>

- [HJK04] Heß, A., Johnston, E. & Kushmerick, N. [ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services](#), *3rd International Semantic Web Conference (ISWC 2004)*, [Lecture Notes in Computer Science](#), Springer Verlag.
- [METEOR-S] <http://lsdis.cs.uga.edu/projects/METEOR-S>
- [ODM] Ontology Definition Metamodel. Revised Submission to OMG/RFP ad/2003-03-40, Jan. 2005, <http://www.omg.org/docs/ad/05-01-01.pdf>
- [OWL] OWL Web Ontology Language Overview, <http://w3.org/TR/owl-features/>
- [OWL-S] Web Ontology Language for Web Services, <http://www.daml.org/services>
- [POSV04] Patil, A., Oundhakar, S., Sheth, A., and Verma, K. [METEOR-S Web service Annotation Framework](#), *Proceeding of the World Wide Web Conference*, New York, NY, May 2004, pp. 553-562.
- [SCL] Delugach, H. ISO/IEC WD 24707 Information technology – Common Logic (CL) – A Framework for a Family of Logic-Based Languages, Pacific Northwest National Laboratory, Chantilly, VA 7 June 2004.
- [S03] 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. [presentation](#)
- [SVSM03] Shivashanmugam K., Verma K., Sheth A., Miller J.. [Adding Semantics to Web Services Standards](#). In the proceedings of *The 2003 International Conference on Web Services (ICWS'03)*, Las Vegas, NV, June 23 - 26, 2003., pp. 395-401. <http://lsdis.cs.uga.edu/library/download/SVSM03-ICWS-final.pdf>
- [SMSV05] Shivashanmugam, K., Miller, J. A., Sheth A., and Verma K. [Framework for Semantic Web Process Composition](#), *International Journal of Electronic Commerce (IJEC)*, Special Issue on Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce, Winter 04/05 issue, 9 (2), pp. 71-106.
- [SPAS03] Sycara, K., Paolucci, M., Ankolekar, A., and Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services, *Journal of Web Semantics*, Volume 1, Issue 1, December 2003
- [TM] Topic Maps Committee. <http://www.topicmaps.org/>
- [WSDL] W3C Note, Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [MVR+04] Miller, J., Verma, K., Rajasekaran, P., [Sheth, A.](#), Aggarwal, R., and Shivashanmugam, K. WSDL-S: A proposal to the W3C WSDL Committee, August 15, 2004 <http://lsdis.cs.uga.edu/projects/WSDL-S/wsdl-s.pdf>
- [WSMO] Web Services Modeling Ontology, <http://www.wsmo.org>
- [WSML] WSML Committee. Web Service Modeling Language 2004, <http://www.wsmo.org/wsml/>