Semantically Annotating a Web Service

Kunal Verma • *Accenture Technology Labs*
Amit Sheth • *Wright State University*

In the past few years, service-oriented architecture (SOA) has transitioned from a partially formed vision into a widely implemented paradigm, with Web services (WS) being the forerunners to implementing SOA-based solutions. But even though the current trend is to use Web services' standards-based nature to establish static connections between various components, businesses are starting to explore dynamic value-added propositions, such as reuse, interoperability, and agility.

Adding Semantics

The building blocks of SOA-based solutions are self-describing Web services that can be reused across various applications. The Web Service Description Language (WSDL) was created specifically for this purpose. It provides several useful constructs for describing services, including

- operation constructs to describe service methods, provide information about parameters, and define types of operations (synchronous, asynchronous);
- operation parameter descriptions via the XML schema; and
- information about the type of protocol needed to invoke the service (such as SOAP over HTTP).

To understand and use a service provider service, a client must understand the semantics of each service operation. In other words, the client must be able to unambiguously decipher each operation's intended purpose as well as the intended content of all elements of its parameters. The service providers and clients have three potential options for addressing this issue:

- *Pre-agreement on all terms for operation names and parameters between service providers and clients*. This approach requires a manual agreement between the service provider and potential clients before they can access the service.
- *Comments for all aspects of a service*. This approach lets clients read comments to understand how to use a particular service, which is analogous to the information semantics supported in electronic data-interchange (EDI) formats. With EDI, interoperability is based on how well application developers understand textual descriptions.
- *Service elements annotated with terms from domain models including industry standards, vocabularies, taxonomies, and ontologies*. This approach negates the need to create agreements with all potential clients and alleviates or eliminates terminological discrepancy. Moreover, if the service provider uses a formal modeling language for annotating the services, machines can process the annotations and ease the human effort required to determine the service's use.

In 2002, researchers proposed Semantic Web services that took the third approach in two different ways. One path represented a revolutionary rethinking of all aspects of semantic services — the Web Ontology Language (OWL-S; www.daml.org/services/owl-s/) and Web Service Modeling Ontology (WSMO; www.wsmo.org) are excellent examples. The other path took a more evolutionary approach that stayed consistent with existing standards and industrial practices. This was exemplified by the managing end-to-end operations for Semantic Web services and processes project (METEOR-S; http://lsdis.cs.uga.edu/projects/meteor-s/). (In this article, the term "we" refers to both authors' research work, with either of us as coauthors, done primarily as part of the METEOR-S.)

 1089-7801/07/$25.00 © 2007 IEEE

Tools and Use Cases

Tools for Semantic Annotation of Web Services (SAWSDL) are available for download at http://lsdis.cs.uga.edu/projects/meteor-s/SAWSDL/. The primary tools include:

- SAWSDL4J is an implementation of the SAWSDL specification that lets developers create SAWSDL-based services.
- Radiant is an Eclipse plug-in for creating and publishing SAWSDL and WSDL-S service interfaces. Users can add annotations to existing service descriptions in the Web Services Description Language using Radiant's GUI.
- Lumina is an Eclipse plug-in for semantic publications and Web service discovery (it enables discovery based on SAWSDL's semantic annotations).

IBM also has some semantic Web service tools (www.alphaworks.ibm.com/tech/wssem) that leverage semantic annotations in SAWSDL for interface matching, discovery, and composition. SAWSDL has several use cases available, but one of the best is a life sciences demonstration at the Stargate Glycomics Web portal (http://128.192.9.86/stargate/index.jsp).

A seminary article we published in 2003 outlined the second approach,[1] describing how to use WSDL's extensibility elements to provide hooks for semantically annotating various service elements. In 2004, we created a more mature version of this work and called it WSDL-S before collaborating with IBM in 2005 to submit a revised and refined WSDL-S specification as a W3C member submission.[2] In 2006, we achieved success — the W3C created a charter for the Semantic Annotation of Web Services (SAWSDL; www.w3.org/2002/ws/sawsdl), which used WSDL-S as its primary input. SAWSDL became a W3C candidate recommendation in January 2007. (See the "Tools and Use Cases" sidebar for details about using SAWSDL.)

How Will Semantics Change SOA?

Several value propositions have pushed enterprises toward implementing SOA-based systems, including the potential for greater reuse of services, the ease of interoperability among SOA services, and greater agility in business processes. However, achieving these benefits puts a burden on specifications. To address this challenge, the METEOR-S project proposed and investigated a broad framework with four types of semantics:

- *Functional semantics*. A formal description of the service's functionality is crucial for efficient service discovery and reuse.[3]
- *Data semantics*. A formal description of the data the service exchanges is crucial for interoperability.[4]
- *Nonfunctional semantics*. Formally defined service-level agreements and quality-of-service attributes are crucial to service providers' efforts to differentiate themselves from their competitors.[5,6]
- *Execution semantics*. Formally modeling the Web service's runtime behavior and exceptions is crucial for ensuring that services execute correctly and for supporting runtime exceptions.[7]

Currently, SAWSDL has direct support for functional and data semantics, but service providers can incorporate nonfunctional and execution semantics with the WS-Policy framework (www.w3.org/2002/ws/policy/). As part of the 2006 Semantic Web Services Challenge (http://sws-challenge.org), we demonstrated how SAWSDL — with different types of semantics — can help model and successfully implement a fairly complex supply scenario.

SAWSDL's Benefits

Canonical data models are excellent for supporting interoperability within and among enterprises. Because many applications already have their own schema, application developers often creates mappings between application-specific schema and canonical data models for interoperability. In SAWSDL, the `modelReference` and `schemaMapping` attributes are the most appropriate for this task. Fortunately, SAWSDL is agnostic to both the domain model and the mapping language, which gives it a lot of flexibility: domain models can be as simple as agreed upon English-language terms or as complex as expressive ontologies that use formal models such as description logics. SAWSDL service providers can choose different mapping languages, based on the modeling paradigm used.

A critical problem for enterprises is finding services that they can reuse across entire business processes. A functional description of the service — along with a discovery mechanism that can leverage the description — is the key to increasing reuse in enterprises. SAWSDL's `modelReference` attribute can link operations to functional descriptions in domain models — for example, the service provider can annotate all operations that implement the Partner Interface Process (PIP) `RequestPurchaseOrder` from the RosettaNet standard with an identifier representing the term. Using a discovery mechanism such as the one proposed in METEOR-S Web Services Discovery Infrastructure (MWSDI),[8] clients can locate all services in the enterprise that implement the `RequestPurchaseOrder` PIP and reuse them as required.

The METEOR-S project has also investigated adding semantics to other Web service standards, such as Web Services Business Process Execution Language (WS-BPEL). A recent study proposed augmenting BPEL descriptions of processes with semantic templates to achieve runtime binding of services based on semantic discovery.[9] Each semantic template would use a SAWSDL description to depict the abstract functionality that a particular service partner required. The process analysts would then use the semantic templates to find and bind partner services to the process at runtime. We demonstrated this approach's utility in a supply-chain scenario, in which we created a supply-chain process using a WS-Process; SAWSDL descriptions captured the required suppliers' semantic templates. Our system was then able to choose optimal suppliers for each part at runtime. This work also showed how using SAWSDL descriptions of services along with the WS-Policy standard can help model a service's runtime execution and exception behavior, and how that model can, in turn, adapt the process to logical exceptions such as delays in ordered goods.

SAWSDL is the first step to infusing semantics into services and SOA. With early tools and use cases already available, we hope to see its near-term impact in better supporting data mediation when services need to interoperate, as in a composition. We also anticipate that the current version of SAWSDL will be enhanced with the ability to model preconditions, post-conditions, and effects as other research efforts in Semantic Web services have explored. In the medium term, we advocate using semantics to improve SOA by enriching policy or agreement specifications (and corresponding W3C proposals and drafts related to WS-Policy and WS-Agreement) that can lead to better partner selection and improved dynamic and adaptive capabilities of services and processes. In the long term, we hope to see a pervasive impact of semantics through all the states of service and process life cycle, encompassing publication, discovery, orchestration, composition, dynamic configuration, and so on, ultimately leading to adaptive Web services and processes.[10]

References

1. K. Sivashanmugam et al., "Adding Semantics to Web Services Standards," *Proc. 3rd IEEE Int'l Conf. Web Services*, IEEE CS Press, 2003, pp. 395–401.
2. R. Akkiraju et al., "Web Service Semantics: WSDL-S," W3C member submission, 2005; www.w3.org/Submission/WSDL-S/.
3. K. Verma et al., "Meteor-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services," *J. Information Technology and Management*, vol. 6, no. 1, 2005, pp. 17–39.
4. M. Nagarajan et al., "Semantic Interoperability of Web Services: Challenges and Experiences," *Proc. 4th IEEE Int'l Conf. Web Services*, IEEE CS Press, 2006, pp. 373–382.
5. J. Cardoso et al., "Quality of Service for Workflows and Web Service Processes," *J. Web Semantics*, vol. 1, no. 3, 2004, pp. 281–308.
6. N. Oldham et al., "Semantic WS-Agreement Partner Selection," *Proc. 15th Int'l World Wide Web Conf.* (WWW2006), ACM Press, 2006, pp. 697–706
7. K. Verma et al., "Optimal Adaptation in Web Processes with Coordination Constraints," *Proc. 4th IEEE Int'l Conf. Web Services*, IEEE CS Press, 2006, pp. 257–264.
8. K. Verma et al., "Meteor-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services," *J. Information Technology and Management*, vol. 6, no. 1, 2005, pp. 17–39.
9. K. Verma, *Configuration and Adaptation of Semantic Web Processes*, PhD thesis, Dept. of Computer Science, Univ. of Georgia, Aug. 2006; http://lsdis.cs.uga.edu/~kunal/verma_kunal_200608_phd.pdf.
10. K. Verma and A. Sheth, "Autonomic Web Processes," *Proc. 3rd Int'l Conf. Service-Oriented Computing (ICSOC), LNCS 3826*, B. Benatallah et al., eds., Springer Verlag, 2005, pp. 1–11.

Kunal Verma is a senior research specialist at Accenture Technology Labs. His research interests include SOA, the Semantic Web, Semantic Web services, and adaptive Web processes. Contact him at k.verma@accenture.com.

Amit Sheth is the LexisNexis Ohio Eminent Scholar and director of the Knowledge-Enabled Information and Services Center (http://knoesis.org) at Wright State University. His research interests include the Semantic Web, services science, and information integration and analysis. Sheth is a fellow of the IEEE. Contact him at amitpsheth@ieee.org.