Web Service Modeling Ontology (WSMO) - An Ontology for Semantic Web Services

Position paper at the W3C Workshop on Frameworks for Semantics in Web Services, June 9-10, 2005, Innsbruck, Austria

Prepared on behalf of the WSMO Working Group by: John Domingue, Dumitru Roman, and Michael Stollberg

The list of the WSMO Working Group members can be found at: http://www.wsmo.org/people.html

The WSMO Working Group Co-Chairs are: Christoph Bussler, John Domingue, and Dieter Fensel

Abstract

This paper outlines some of the main issues related to the semantic modeling of Web Services and provides an overview of the Web Service Modeling Ontology (WSMO) - an ontology for Semantic Web Services. The design principles of this ontology are highlighted and a short description of the top-level elements is given. The conceptual model summarized in this paper represents the foundation for Semantic Web Services from the viewpoint of the Web Service Modeling Ontology (WSMO) Working Group.

Table of Contents

---

1. Introduction and Motivation

The Semantic Web and Web Services are envisioned as the enabling technologies for next the generation of web applications. The former aims at enhancing the machine-readability of web content, whereof ontologies have been identified as the key technical building block. The objective of Web Services is to enable distributed computation over the Internet by automated and dynamic discovery, composition, and execution of services, thus providing a new technology for web-based system engineering. The current Web Service technology stack enables the exchange of messages between Web Services (SOAP), describes the technical interface for consuming a Web Service (WSDL), and supports the advertisement of Web Services in registries (UDDI). However, these technologies do not include explicit descriptions of the functionality of a Web Service. Moreover, the existing descriptions are represented syntactically and therefore do not depict the meaning of the information to be interchanged. Semantic Web Services (SWS) applies Semantic Web technology to Web Services raising the level of discourse. More specifically, through the use of exhaustive semantic description frameworks SWS will support the provision of intelligent mechanisms for the discovery, composition, contracting, and execution of Web Services.

In this context, this position paper presents the Web Service Modeling Ontology (WSMO) as an ontology for semantically describing Semantic Web Services. Taking the Web Service Modeling Framework (WSMF) [Fensel & Bussler, 2002] as a starting point, WSMO refines and extends this framework, and develops a formal ontology and language. WSMF consists of four different main elements for describing semantic Web Services: (1) ontologies which provide the concepts and relationships used by other elements, (2) goals that define the users' objectives, i.e. the (potential) problems that should be solved by Web Services, (3) Web Services descriptions that define various aspects of a Web Service, and (4) mediators which bypass interoperability problems.

The Web Service Modeling Ontology (WSMO) is developed in the context of WSMO Working Group, as part of the SDK cluster, with the aim of, through alignment between key European research projects in the Semantic Web Service area, the further the development of Semantic Web Services and works toward further standardization in the area of Semantic Web Service languages and to work toward a common architecture and platform for Semantic Web Services. The WSMO Working Group includes the WSML Working Group, which aims at developing a language called Web Service Modeling Language (WSML) that formalizes the Web Service Modeling Ontology (WSMO), and the WSMX Working Group, which aims at providing an execution environment and a reference implementation for WSMO.

The rest of this paper is organized as follows: Section 2 presents the WSMO's design principles, Section 3 gives a high level overview of the ontology by presenting its top-level elements, Section 4 relates WSMO to other initiatives in this area, and Section 5 concludes the position paper.

2. WSMO Design Principles

WSMO provides ontological specifications for the core elements of Semantic Web Services. In fact, Semantic Web Services aim at an integrated technology for the next generation of the Web by combining Semantic Web technologies and Web Services, thereby turning the Internet from a information repository for human consumption into a world-wide system for distributed web computing. Therefore, appropriate frameworks for Semantic Web Services need to integrate the basic Web design principles, those defined for the Semantic Web, as well as design principles for distributed, service-orientated computing of the Web. WSMO is therefore based on the following design principles:

- **Web Compliance** - WSMO inherits the concept of URI (Universal Resource Identifier) [URI] for unique identification of resources as the essential design principle of the Word Wide Web. Moreover, WSMO adopts the concept of Namespaces for denoting consistent information spaces, supports XML and other W3C Web technology recommendations, as well as the decentralization of resources.
- **Ontology-Based** - Ontologies are used as the data model throughout WSMO, meaning that all resource descriptions as well as all data interchanged during service usage are based on ontologies. Ontologies are a widely accepted state-of-the-art knowledge representation, and have thus been identified as the central enabling technology for the Semantic Web. The extensive usage of ontologies allows semantically enhanced information processing as well as support for interoperability; WSMO also supports the ontology languages defined for the Semantic Web.
- **Strict Decoupling** - Decoupling denotes that WSMO resources are defined in isolation, meaning that each resource is specified independently without regard to possible usage or interactions with other resources. This complies with the open and distributed nature of the Web.
- **Centrality of Mediation** - As a complementary design principle to strict decoupling, mediation addresses the handling of heterogeneities that naturally arise in open environments. Heterogeneity can occur in terms of data, underlying ontology, protocol or process. WSMO recognizes the importance of mediation for the successful deployment of Web Services by making mediation a first class component of the framework.
- **Ontological Role Separation** - Users, or more generally clients, exist in specific contexts which will not be the same as for available Web Services. For example, a user may wish to book a holiday according to preferences for weather, culture and childcare, whereas Web Services will typically cover airline travel and hotel availability. The underlying epistemology of WSMO differentiates between the desires of users or clients and available services.
- **Description versus Implementation** - WSMO differentiates between the descriptions of Semantic Web Services elements (description) and executable technologies (implementation). While the former requires a concise and sound description framework based on appropriate formalisms in order to provide a concise for semantic descriptions, the latter is concerned with the support of existing and emerging execution technologies for the Semantic Web and Web Services. WSMO aims at providing an appropriate ontological description model, and to be complaint with existing and emerging technologies.

- **Execution Semantics** - In order to verify the WSMO specification, the formal execution semantics of reference implementations like WSMX as well as other WSMO-enabled systems provide the technical realization of WSMO.

3. WSMO Top-level Elements

The following briefly outlines the conceptual model of WSMO – the complete specification can be found in [Roman et al., 2005]. The elements of the WSMO ontology are defined in a meta-meta- model language based on the Meta Object Facility (MOF) [MOF]. In order to allow complete item descriptions, every WSMO element is described by non-functional properties. These are based on the Dublin Core (DC) Metadata Set [DublinCore] for generic information item descriptions, and other service-specific properties related to the quality of service.

Ontologies. Ontologies provide the formal semantics for the terminology used within all other WSMO components. Using MOF, we define an ontology as described in the Listing 1 below:

Listing 1. Ontology Definition

```
Class ontology
    hasNonFunctionalProperties type nonFunctionalProperties
    importsOntology type ontology
    usesMediator type ooMediator
    hasConcept type concept
    hasRelation type relation
    hasFunction type function
    hasInstance type instance
    hasAxiom type axiom
```

A set of *non-functional properties* are available for characterizing ontologies; they usually include the DC Metadata elements. *Imported ontologies* allow a modular approach for ontology design and can be used as long as no conflicts need to be resolved between the ontologies. When importing ontologies in realistic scenarios, some steps for aligning, merging and transforming imported ontologies in order to resolve ontology mismatches are needed. For this reason *ontology mediators* are used (ooMediators). *Concepts* constitute the basic elements of the agreed terminology for some problem domain. *Relations* are used in order to model interdependencies between several concepts (respectively instances of these concepts); *functions* are special relations, with a unary range and a n-ary domain (parameters inherited from relation), where the range value is functionally dependent on the domain values, and *instances* are either defined explicitly or by a link to an instance store, i.e., an external storage of instances and their values.

Web Services. WSMO provides service descriptions for describing services that are requested by service requesters, provided by service providers, and agreed between service providers and requesters. In the Listing 2 below, the common elements of these descriptions are presented.

Listing 2. Service Description Definition

```
Class service
    hasNonFunctionalProperties type nonFunctionalProperties
    importsOntology type ontology
    usesMediator type {ooMediator, wwMediator}
    hasCapability type capability multiplicity = single-valued
    hasInterface type interface
```

Within the service class the *non-functional properties* and *imported ontologies* attributes play a role that is similar to that found in the ontology class with the minor addition of a quality of service non-functional property. An extra type of *mediator* to deal with protocol and process related mismatches between web services is also included.

The final two attributes define the two core WSMO notions for semantically describing Web Services: a *capability* which is a functional description of a Web Service, describing constraints on the input and output of a service through the notions of preconditions, assumptions, postconditions, and effects; and *service interfaces* which specify how the service behaves in order to achieve its functionality. A service interface consists of a *choreography* which describes the interface for the client-service interaction required for service consumption, and an *orchestration* which describes how the functionality of a Web Service is achieved by aggregating other Web Services.

Goals. A goal specifies the objectives that a client may have when consulting a Web Service, describing aspects related to user desires with respect to the requested functionality and behavior. Ontologies are used as the semantically defined terminology for goal specification. Goals model the user view in the Web Service usage process and therefore are a separate top level entity in WSMO.

Listing 3. Goal Definition

```
Class goal
    hasNonFunctionalProperties type nonFunctionalProperties
    importsOntology type ontology
    usesMediator type {ooMediator, ggMediator}
    requestsCapability type capability multiplicity = single-valued
    requestsInterface type interface
```

As presented in Listing 3 above, the *requested capability* in the definition of a goal represents the functionality of the services the user would like to have, and the *requested interface* represents the interface of the service the user would like to have and interact with.

Mediators. The concept of Mediation in WSMO addresses the handling of heterogeneities occurring between elements that shall interoperate by resolving mismatches between different used terminologies (data level), on communicative behavior between services (protocol level), and on the business process level. A WSMO Mediator connects elements and provides mediation facilities for resolving mismatches. The description elements of a WSMO Mediator are its source and target elements, and the mediation service for resolving mismatches, as shown in the Listing 4 below.

Listing 4. Mediators Definition

```
Class mediator
    hasNonFunctionalProperties type nonFunctionalProperties
    importsOntology type ontology
    hasSource type {ontology, goal, service, mediator}
    hasTarget type {ontology, goal, service, mediator}
    hasMediationService type {goal, service, wwMediator}
```

WSMO defines different types of mediators for connecting the distinct WSMO elements: *OO Mediators* connect and mediate heterogeneous ontologies, *GG Mediators* connect Goals, *WG Mediators* link Web Services to Goals, and *WW Mediators* connects interoperating Web Services resolving mismatches between them.

4. Related Work

OWL-S is an ontology for describing Web Services represented in OWL and as such is comprised of three top-level notions [OWL-S]: the *Service Profile* includes information for 'service advertisement' which is used for Web Service Discovery; the *Service Model* contains descriptive information on the functionality of a service and its composition out of other services, whereby the service functionality is conceived as a process; the *Service Grounding* gives details of how to access the service, mapping from an abstract to a concrete specification for service usage. At the epistemological level the differences between OWL-S and WSMO are related to the principles of Ontological Role Separation and Centrality of Mediation. Specifically, that WSMO contains the top level notions of Goal and Mediator. Other differences are that WSMO uses WSML [de Bruijn., 2005] a semantic web language targeted specifically at SWS and WSMO has a reference implementation WSMX.

METEOR-S aims at integrating web service technologies such as Web Services Business Process Execution Language (WSBPEL), Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) with Semantic Web technologies in order to automate the tasks of publication, discovery, description, and control flow of web services. Compared to WSMO, METEOR-S follows a much more technology centered approach, not providing a conceptual model for the description of services and their related aspects.

IRS III [Domingue et al., 2004] a framework and implemented infrastructure which supports the creation of semantic web services according to the WSMO ontology. IRS III has four main classes of features which distinguish it from other work on semantic web services. Firstly, it supports *one-click publishing* of 'standard' programming code. In other words, it automatically transforms programming code (currently it supports Java and Lisp environments) into a web service, by automatically creating the appropriate wrapper. Hence, it is very easy to make existing standalone software available on the net, as web services. Secondly, by extending the WSMO goal and web service concepts users of IRS III directly invoke web services via goals i.e. IRS III supports *capability-driven* service execution. Thirdly, IRS III is programmable. IRS III users can substitute their own semantic web services for some of the main IRS III components. Finally, IRS

III services are web service compatible – standard web services can be trivially published through the IRS III and any IRS III service automatically appears as a standard web service to other web service infrastructures.

5. Conclusions

Semantic Web Services are one of the most promising research directions to improve the integration of applications within and across enterprise boundaries. In this context, WSMO has the aim of providing the conceptual and technical means to realize Semantic Web Services, improving the cost-effectiveness, scalability and robustness of current solutions. The ontology highlighted in this position paper provides the core elements that are needed to represent semantic web services and related issues: ontologies, that provide the common terminology used by other WSMO elements, services that are requested, provided, and agreed upon by requesters and providers, goals that are description of problems that should be solved by services, and mediators, which deal with interoperability problems between different WSMO elements.

In total, we believe that our framework consisting of an ontology and the language for describing web services semantically sets a solid basis for attacking the research issues associated with Semantic Web Services. A document that explains WSMO in more depth can be found in the WSMO Primer [Feier, 2005]. A set of WSMO tutorials, presented at different international conferences (AIMSA 2004, Net Object Days 2004, ISWC 2004) and project meetings (WSMO training for DIP and ASG), that provide a detailed presentation of WSMO, can be can be found in [Stollberg & Arroyo, 2005]. Two of the presentations are available as webcasts at http://stadium.open.ac.uk/dip/.

Several use cases demonstrating of how to use WSMO in a real-world settings can be found in the WSMO Use Case Modeling and Testing documents [Stollberg et al., 2004]. Additionally, within the EU funded project COCOON project [Valle et al., 2004] WSMO is currently being applied to a health application, and within the DIP Integrated Project WSMO is being deployed in the domains of eGovernment, eBanking and telecommunications. For the different subsets of the language for defining WSMO annotated services we refer the reader to the WSML Family of Representation Languages [de Bruijn, 2004], and a logical framework for service discovery in WSMO has been defined in [Keller et al., 2005].

In addition to the theoretical results and deployment scenarios a number of WSMO a number of compliant tools have been developed or are currently under development. These include: WSMX - an execution environment for dynamic matchmaking, selection, mediation and invocation of semantic web services based on WSMO, IRS III - a platform and infrastructure for creating WSMO-based Semantic Web Services, SWWS Studio and the WSMO Studio - WSMO compliant editors, and WSMO4J - an API and a reference implementation for building Semantic Web Services applications compliant with WSMO in Java.

Acknowledgments

The authors would like to thank to all the members of the WSMO, WSML, and WSMX Working Groups for their advice and input into this document.

References

[de Bruijn., 2005] de Bruijn, J. (Ed.): *The WSML Specification*, WSML Deliverable D16, WSML Working Draft, 2005, latest version available at http://www.wsmo.org/TR/d16/.

[Domingue et al., 2004] Domingue, J.; Cabral, L.; Hakimpour, F.; Sell D.; Motta, E. (2004). IRS III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. *Proceedings of the Workshop on WSMO Implementations (WIW 2004)* Frankfurt, Germany, September 29-30, 2004, CEUR Workshop Proceedings, ISSN 1613-0073. Available from http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-113/paper3.pdf.

[DublinCore] Weibel S.; Kunze, J.; Lagoze,C.; Wolf, M.: *RFC 2413 - Dublin Core Metadata for Resource Discover*y, September 1998, available at http://www.ietf.org/rfc/rfc2413.txt.

[Feier, 2004] Feier, C. (Ed.): *WSMO Primer*, WSMO Deliverable D3.1, WSMO Working Draft, 2005, latest version available at http://www.wsmo.org/2004/d3/d3.1/.

[Fensel & Bussler, 2002] Fensel,D.; Bussler, C.: *The Web Service Modeling Framework WSMF*, Electronic Commerce Research and Applications, 1(2), 2002.

[Keller et. al., 2004] Keller, U.; Lara, R.; Polleres, A. (Eds.): *WSMO Web Service Discovery*, WSMO Deliverable D5.1, 2004, latest version available at http://www.wsmo.org/2004/d5/d5.1/v0.1/

[MOF] The Object Management Group: Meta-Object Facility, version 1.4, 2002. Available at http://www.omg.org/technology/documents/formal/mof.htm.

[OWL-S] The OWL Services Coalition. *OWL-S 1.1 beta release*. Available at http://www.daml.org/services/owl-s/1.1B/, July 2004.

[Roman et al., 2005] Roman, D.; Lausen, H.; Keller, U. (Eds.): The Web Service Modeling Ontology WSMO, final version 1.1. WSMO Final Draft D2, 2005,latest version available at http://www.wsmo.org/TR/d2/v1.1/.

[Stollberg & Arroyo, 2005] Stollberg, M.; Arroyo S. (Eds.) *WSMO Tutorial*. WSMO Deliverable D17, WSMO Working Draft, 2005, latest version available at http://www.wsmo.org/TR/d17/.

[Stollberg et al., 2004] Stollberg, M.; Lausen H.; Polleres, A.; Lara,R. (Eds.): *WSMO Use Case Modeling and Testing*, WSMO Deliverable D3.2, WSMO Working Draft, 2004, latest version available at http://www.wsmo.org/2004/d3/d3.2/

[URI] Berners-Lee, T.; Fielding, R.; Masinter, L.: *RFC 3986 - Uniform Resource Identifiers (URI): Generic Syntax*, IETF, January 2005, available at http://www.isi.edu/in-notes/rfc3986.txt.

[Valle et al., 2004] Della Valle, E.; Cerizza, D.; Gadda, L. (Eds.). *Use Case: Semantic Discovery of Community of Practice*. 2004. COCOON Working Draft, December 2004, available at http://cocoon.cefriel.it/RD2/usecases/semantic-discovery-of-cop.