



D30v0.1 Aligning WSMO and WSDL-S

WSMO Working Draft 5 August 2005

This version:

<http://www.wsmo.org/TR/d30/v0.1/20050805/>

Latest version:

<http://www.wsmo.org/TR/d30/v0.1/>

Previous version:

none

Authors:

Jacek Kopecký

This document is also available in non-normative [PDF](#) version.

Copyright © 2005 DERI®, All Rights Reserved. DERI liability, trademark, document use, and software licensing rules apply.

Table of contents

- 1. Introduction
- 2. Brief summary of WSDL-S
- 3. Aligning WSDL-S with WSMO
 - 3.1 Analyzing the coverage of WSMO in WSDL-S
 - 3.2 Proposed extensions to WSDL-S
- 4. Using WSDL-S for WSMO grounding
- 5. Conclusions and Further Work
- References
- Acknowledgement

1. Introduction

Web Service Semantics ([WSDL-S]) is a set of elements extending WSDL [WSDL] to allow semantic annotation of WSDL descriptions. WSDL-S is independent of any particular semantic annotation language, examples of which include WSMO ([WSMO]) and OWL-S ([OWL-S]). This document is a proposal for extending WSDL-S so that it usefully covers the features of WSMO, without compromising the main goals of WSDL-S: simplicity and independence.

In [section 2](#) below, we briefly summarize WSDL-S. In the following [section 3](#) we analyze how much of WSMO can already be referenced in WSDL-S and propose extensions for fuller coverage. Finally in [section 4](#) we discuss how WSDL-S with these extensions could be used for WSMO grounding.

2. Brief summary of WSDL-S

In summary, WSDL-S is a set of semantic annotation functionalities for WSDL files. It chooses to base semantic annotations within WSDL files "in an upwardly compatible manner so as not to disrupt the existing install-base of Web services."

WSDL-S defines a *Semantic Model* to capture the terms and concepts used to describe and represent an area of knowledge, in our case a Web service. Four particular parts of the semantic model are distinguished:

- Input Semantics – the meaning of input parameters
- Output Semantics – the meaning of output parameters
- Precondition – a set of semantic statements that are required to be true before an operation can be successfully invoked
- Effect – a set of semantic statements that must be true after an operation completes execution after being invoked. Different effects can be true depending on whether the operation completed successfully or unsuccessfully.

WSDL-S uses the extensibility of WSDL and provides five elements and attributes which, when used in the appropriate parts of a WSDL document, provide hooks for attaching semantics to WSDL components. The five elements and attributes are described in the list below:

- **modelReference** attribute, whose URI value points to a concept in the semantic model of the Web service and specifies that there is a one-to-one mapping between the owner of the attribute and the referenced concept. This attribute can be used by itself on an XML Schema element or type declaration to link directly to an ontology concept, or it can be used as part of specifying a precondition or effect as described further below.
- **schemaMapping** attribute, whose URI value points to a mapping between a schema and ontology concepts. This attribute is to be used on an XML Schema element or type declaration and can point to a transformation, for example in XSLT, from XML data that follows this schema to ontological data, for example in RDF/OWL or in WSML.
- **precondition** and **effect** are similarly structured elements that are used on WSDL interface operations to specify conditions that must hold before and after the operation is invoked. The conditions are specified either by reference to the semantic model (reusing the *modelReference* attribute) or directly by writing the condition as the value of the *expression* attribute of the two elements.
- **category** element provides a pointer to some taxonomy category. It can be used on a WSDL interface and is intended to be used for taxonomy-based discovery or when the WSDL interface is filed in a repository.

3. Aligning WSDL-S with WSMO

This section first analyzes how much of WSMO can already be referenced with WSDL-S and then proposes how fuller coverage of WSMO could be achieved. In particular, [section 3.1](#) compares what WSMO can express with what WSDL-S can reference and [section 3.2](#) enumerates the WSDL-S extensions proposed for full coverage of WSMO in WSDL-S.

3.1 Analyzing the coverage of WSMO in WSDL-S

In WSDL-S, the semantic model of a Web service is expected to contain the semantics of input and output parameters and the specifications of preconditions and effects of service operations, plus the categorization of a WSDL interface. WSMO specifies a more detailed model where a Web service can have a capability with preconditions, assumptions, postconditions and effects; and an interface with choreography and orchestration; and where data is described using ontologies.

The semantics of input and output parameters in WSDL-S are indicated with annotations in the XML Schema used in the WSDL document. These annotations link elements and types with semantic model (ontology) concepts. In WSMO Grounding [[WSMOGrounding](#)], the semantics of inputs and outputs are currently specified by linking choreography (see [[WSMOChoreography](#)]) input and output concepts to WSDL messages, which correspond to XML Schema elements. In short, both WSDL-S and WSMO link schema elements with ontology concepts. So annotating data is covered in WSDL-S.

Categorization of a WSDL interface in WSDL-S can be compared to the non-functional properties *Subject* and *Type* on a WSMO Web service, therefore it can be assumed that here, too, WSDL-S model covers WSMO well.

Finally, WSDL-S is based on the WSDL model of Web services interfaces consisting of separate operations, and preconditions and effects in WSDL-S are attached to operations. In contrast, WSMO distinguishes between preconditions, assumptions, postconditions and effects, but these aspects are modeled on the whole Web service (inside its capability, to be precise), because WSMO talks about a Web service as a whole, without splitting it to operations. Additionally, WSMO models the choreography and orchestration interface(s) of a Web service, an aspect not covered by WSDL-S at all. These are the differences that need to be resolved to allow a useful combination of WSDL-S with

WSMO. A particular solution is suggested in the following subsection.

3.2 Proposed extensions to WSDL-S

In the previous section, three areas were identified where WSDL-S doesn't fully cover the capabilities of WSMO: semantic conditions on the execution of a whole Web service (as opposed to a single operation), and references to choreography and orchestration. The following paragraphs define how these gaps could be covered.

As a very small change, we first suggest that WSDL-S *precondition* and *effect* elements be allowed on WSDL interface, not only on particular operations. The resulting WSDL-S interface preconditions could then cover both the WSMO preconditions and assumptions, and the WSDL-S interface effects could cover WSMO postconditions and effects. The distinction between WSMO information-space conditions (preconditions and postconditions) and descriptions of the state of the world (assumptions and effects) could be modeled in the value of the *Scope* or *Type* non-functional properties of the appropriate axioms in WSMO.

In order to refer to a choreography, reflecting constraints on the use of the operations in a WSDL interface, WSDL-S should be extended with a *choreography* element, with structure similar to the postcondition and effect elements. This still falls into scope of WSDL-S, as it describes the abstract interface of a Web service. For use with WSMO, this element would refer to a WSMO choreography. Note, however, that choreography, used as a description of the constraints of the use of the operations, is an alternative to preconditions and effects on operations, which could also be described using WSML logical language and referenced from WSDL-S using the existing mechanisms.

Finally, for full coverage of WSMO, we need to refer to an orchestration. Orchestration is an aspect of the implementation of a Web service, rather than of the public interface between a Web service and its clients, and there can be other implementation-specific aspects described on a WSMO Web services. Therefore we suggest that WSDL-S should link from WSDL service to WSMO Web service, reusing the *modelReference* attribute, because a WSMO Web service is a direct semantic model of a WSDL service. This, though, seems out of scope of WSDL-S, which currently wants to cover the abstract interface only. We do not expect that this extension will necessarily be added to WSDL-S.

4. Using WSDL-S for WSMO grounding

The purpose of a WSMO grounding is to complement the semantic description of a Web service with the concrete details about how communication is done "on the wire". While WSMO contains elements intended to make WSMO independent of any particular grounding, like WSDL-S is independent of any particular semantic language, a WSDL-S document with the extensions proposed above could contain all the information necessary to ground a WSMO description and so it could be used in lieu of those mentioned grounding elements in WSMO. This section details for each WSMO grounding element how the necessary information can be retrieved from a WSDL-S document.

The pertinent grounding elements in WSMO are currently these two:

1. in WSMO choreography, the state signature uses the keyword *withGrounding* to point to a grounding mechanism for input, output and shared concepts,
2. and a Web service can have the non-functional property *endpointDescription* to point to a description of the endpoint address and networking details necessary to exchange messages with the service.

In WSDL terms, input, output and shared concepts are grounded to messages within operations, and *endpointDescription* points to a WSDL service, which specifies the endpoint URI(s) and the appropriate network binding(s).

In WSDL-S, schema elements refer to their semantic equivalents (like ontology concepts) using *modelReferences* and *schemaMappings*. Since each WSDL operation message points to one schema element, establishing the reverse mappings from ontology concepts to operation messages, necessary for executing a WSMO choreography, is straightforward: for every WSDL-S link from an element to a concept we find which WSDL interface operation uses that element as its message, and if the concept is used as an input or output of the choreography referenced from this interface (using the proposed WSDL-S element *choreography*), the grounding link goes from the concept to the

WSDL interface operation found above.

As for the *endpointDescription* property pointing to the endpoint and binding details, we can safely assume that a user of WSDL-S already knows the WSDL service to invoke and thus has all the necessary information.

5. Conclusions and Further Work

To summarize, we propose the following changes and extensions to WSDL-S:

1. the elements *precondition* and *effect* should be allowed on WSDL interface
2. the element *choreography* should be added for pointing from a WSDL interface to some choreography description
3. the attribute *modelReference* might be allowed on WSDL service

These changes would enable WSDL-S to point to all relevant parts of WSMO, and WSDL-S documents could be used as grounding for WSMO descriptions.

This document is currently a very first draft intended for initial discussion, it can be expected to evolve and change significantly in subsequent revisions.

References

[WSMOChoreography] J. Scicluna, A. Polleres, D. Roman, C. Feier (editors): *Ontology-based Choreography and Orchestration of WSMO Services*, v0.2 available at <http://www.wsmo.org/TR/d14/v0.2/>

[WSMOGrounding] J. Kopecký, M. Moran, D. Roman, A. Mocan: *WSMO Grounding*, version 0.1 available at <http://www.wsmo.org/TR/d24/d24.2/v0.1/>

[OWL-S] D. Martin (editor): *OWL-S: Semantic Markup for Web Services*, version 1.1 available at <http://www.daml.org/services/owl-s/1.1/overview/>

[WSDL] R. Chinnici, J-J. Moreau, A. Ryman, S. Weerawarana (editors): *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*, W3C Working Draft, 10 May 2005, available at <http://www.w3.org/TR/wsd20/>

[WSDL-S] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.T. Schmidt, A. Sheth, K. Verma: *Web Service Semantics - WSDL-S*, available at <http://lstdis.cs.uga.edu/library/download/WSDL-S-V1.html>

[WSMO] D. Roman, H. Lausen, U. Keller (editors): *Web Service Modeling Ontology (WSMO)*, version 1.1 available at <http://www.wsmo.org/TR/d2/v1.1/>

Acknowledgement

The work is funded by the European Commission under the projects DIP, Knowledge Web, InfraWebs, SEKT, SWWS, ASG and Esperanto; by Science Foundation Ireland under the DERI-Lion project; by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the projects RW² and TSC.

The editors would like to thank to all the [members of the WSMO working group](#) for their advice and input to this document.

\$Date: 2005/08/05 20:12:18 \$

[webmaster](#)