



D24.1v0.1 Aligning WSMO with existing Web services specifications

WSMO Working Draft 16 February 2007

This version:

<http://www.wsmo.org/TR/d24/d24.1/v0.1/20070216/>

Latest version:

<http://www.wsmo.org/TR/d24/d24.1/v0.1/>

Previous version:

<http://www.wsmo.org/2005/d24/d24.1/v0.1/20070112/>

Authors:

Jacek Kopecký
Dumitru Roman

This document is also available in non-normative PDF version.

Copyright © 2007 DERI®, All Rights Reserved. DERI liability, trademark, document use, and software licensing rules apply.

Table of contents

- 1. Introduction
 - 1.1 Relevant Web Services Specifications
 - 2. Web Services Addressing
 - 2.1 WS Addressing Overview
 - 2.2 WS Addressing and WSMO
 - 2.3 Ontology for Endpoint References
 - 3. Web Services Metadata Exchange
 - 3.1 WS-MEX Overview
 - 3.2 Using WS-MEX to retrieve WSMO descriptions
 - 4. Conclusions
 - References
 - Acknowledgement
 - Change Log
-

1. Introduction

Web Services Modeling Ontology (WSMO, [Roman et al. 2004]) allows semantic description of Web services. The current stack of Web services specifications, on the other hand, consists of purely syntactic languages whose semantics is described in plain text. As WSMO is mostly complementary to the other specifications, it should be able to be deployed and used together with the others. Specifically, some of the Web services specifications may want to carry or reference WSMO descriptions, and similarly WSMO descriptions may usefully include or reference elements from the current Web services specifications.

This document is intended as a listing of those Web services specifications that may use WSMO or be used by WSMO, together with a description of the mechanisms linking WSMO and the particular Web services specifications.

1.1 Relevant Web Services Specifications

Due to the modular nature of Web services Specifications, various standardization bodies, industry alliances or even single vendors have been releasing many specifications relevant to Web services, covering larger or smaller domains and purposes. It would be an enormous task to investigate the possible connection of WSMO with each of the specifications, but luckily we only need to focus on a small subset of the space.

Web services specifications can be broadly categorized as *description languages* and *protocols*. Description languages are used to specify various (generally) static deployment and configuration aspects of Web services. Protocols, on the other hand, specify the run-time behaviors necessary to achieve a specific purpose. As an example of this distinction we can take the domain of Web services security which contains both description languages specifying the formats of security tokens like keys and encrypted data, and protocols specifying how those tokens are exchanged and used to achieve a confidential communication channel.

As WSMO falls into the category of description languages, this document focuses on relating WSMO to other description languages or to protocols that refer to or transfer pieces of descriptions. Initially we focus on standards (finished or under development) and on major finished non-standard industry specifications.

The following is a listing of the Web services specifications that we have identified as potentially related to WSMO:

- Web Services Description Language and XML Schema (see [D24.2 WSMO Grounding](#))
- Web Services Policy ([D24.3 Aligning WSMO with WS-Policy](#))
- Web Services Addressing ([Section 2](#))
- Web Services Metadata Exchange (WS-MEX) protocol ([Section 3](#))
- Universal Description, Discovery and Integration (UDDI) repository
- Web Services Discovery
- Web Services Resource Framework (WSRF)
- Business Process Execution Language for Web Services (BPEL4WS)
- Web Services Choreography Description Language (WS-CDL)
- ebXML

Note that this is an ongoing work, further sections or deliverables may be added in the future to discuss specifications that are not yet discussed here.

2. Web Services Addressing

The Web Services Addressing specification [[WSAddressing](#)] defines a format for references to Web services endpoints (so called endpoint references) and a set of message headers used for simple message routing and correlation purposes. Only the endpoint reference structure is directly relevant to WSMO.

2.1 WS Addressing Overview

The following is the structure of an endpoint reference in a pseudo schema:

```
01 <wsa:EndpointReference>
02   <wsa:Address>xs:anyURI</wsa:Address>
03   <wsa:ReferenceProperties>
04     ...
05   </wsa:ReferenceProperties> ?
06   <wsa:ReferenceParameters>
07     ...
08   </wsa:ReferenceParameters> ?
09   <wsa:PortType>
10     xs:QName
11   </wsa:PortType> ?
12   <wsa:ServiceName PortName="xs:NCName"?>
13     xs:QName
14   </wsa:ServiceName> ?
15   <wsp:Policy>
16     ...
17   </wsp:Policy>*
18 </wsa:EndpointReference>
```

Endpoint references contain several important pieces of data: address and reference properties (lines 2-5), which together uniquely identify the Web service endpoint; reference parameters (lines 6-8) that are associated with the endpoint to facilitate a particular interaction; pointers to WSDL port type and service port (lines 9-14), also known as interface and service endpoint in WSDL 2.0; and finally a policy (lines 15-17).

2.2 WS Addressing and WSMO

When suitable, an endpoint reference can point to a WSDL interface (as in the following example, lines 5-7) so that the recipient of this reference may choose the appropriate code to use that interface. In similar spirit, we introduce element `wsml:serviceReference` that refers to a WSMO service description to be used in situations where the recipient of an endpoint reference is able to tailor its behavior according to the semantic description of the referenced endpoint (see the example, lines 8-10).

```
01 <wsa:EndpointReference>
02   <wsa:Address>
03     http://fabrikam123.example.com/acct
04   </wsa:Address>
05   <wsa:PortType>
06     fabrikam:InventoryPortType
07   </wsa:PortType>
08   <wsml:serviceReference>
09     http://fabrikam123.example.com/wsml/acct
10   </wsml:serviceReference>
11 </wsa:EndpointReference>
```

Regarding using WSMO as part of the policy in an endpoint reference, see deliverable D24.3 Aligning WSMO with WS-Policy.

2.3 Ontology for Endpoint References

As described above, WS Addressing defines the XML structure of an Endpoint Reference. This structure is used in WS Addressing for routing information, but it is intended to be reused by any applications that require need to reference Web services. In certain situations Endpoint References will be directly incorporated into messages passed between a Web service and its client. In these scenarios the schema of the messages will refer to the Endpoint Reference, like in the following example:

```
01 <xsd:element name="ServiceCreationResponse">
02   <xsd:complexType>
03     <xsd:sequence>
04       <xsd:element name="TTL" type="xsd:integer"/>
05       <xsd:element name="Endpoint" type="wsa:EndpointReferenceType"/>
06     </xsd:sequence>
07   </xsd:complexType>
08 </xsd:element>
```

Similarly, Semantic Web services might need to exchange Web service references. For this purpose we define a simple ontology for Endpoint References modeled after WS Addressing:

```
01 ontology _"http://wsmo.org/ontology/WebServices/"
02   concept EndpointReference
03     hasAddress ofType (1) xs:anyURI
04     hasReferenceProperty ofType rdf#XMLLiteral
05     hasReferenceParameter ofType rdf#XMLLiteral
06     isWSDLPortType ofType xs#QName
07     isWSDLService ofType xs#QName
08     isWSDLPort ofType xs#NCName
09     isWSMOService ofType xs:anyURI /* because webService is not a concept */
10   /* policy belongs to non-functional properties */
```

3. Web Services Metadata Exchange

Web services have metadata to describe what their clients need to know to interact with them. Specifically, Web services can have WS Policy [WSPolicy] policies, WSDL [WSDL] descriptions XML Schema [XMLSchema] schemata associated with them. WSMO [Roman et al. 2004] adds another form of Web service metadata - their semantic descriptions.

To bootstrap communication with Web services and to retrieve these and other types of metadata, Web Services Metadata Exchange (WS-MEX) [WSMEX] specification defines a protocol for requesting metadata from Web services or from dedicated metadata endpoints, described shortly in section 3.1. It is natural for us to enable the retrieval of WSMO descriptions using WS-MEX, this is described in section 3.2.

3.1 WS-MEX Overview

From the point of view of WS-MEX, every Web service can have multiple pieces of metadata in differing languages. To retrieve a service's metadata, a requester can use the Get Metadata request. As a part of the request message, the requester can specify the so-called dialect and identifier of the requested piece of metadata. The following illustrates the structure of the Get Metadata request message:

```
01 <soap:Body>
02   <wsx:GetMetadata>
03     [<wsx:Dialect>xs:anyURI</wsx:Dialect>
04     [<wsx:Identifier>xs:anyURI</wsx:Identifier>]?
05     ]?
06   </wsx:GetMetadata>
07 </soap:Body>
```

The response message contains zero or more so called metadata sections, as illustrated below:

```
01 <soap:Body>
02   <wsx:Metadata ...>
03     [<wsx:MetadataSection Dialect='xs:anyURI'
04       [Identifier='xs:anyURI']? >
05       [
06         <dialectSpecificElementName>...</dialectSpecificElementName>+
07         |
08         <wsx:Location>xs:anyURI</wsx:Location>
09         |
10         <wsx:MetadataReference ...>
11         endpoint-reference
```

```

12     </wsx:MetadataReference>
13   ]
14 </wsx:MetadataSection>]*
15 </wsx:Metadata>
16 </soap:Body>

```

Each metadata section uses the attribute Dialect (line 3) to identify the language and version, in which the metadata is represented. Appendix I of WS-MEX defines the dialect URIs for WSDL 1.1, XML Schema 1.0, WS Policy and WS Policy Attachment and finally for WS Metadata Exchange itself so that services can have nested metadata sets. The optional dialect-dependent attribute Identifier (line 4) provides further identification of this piece of metadata, usually its target namespace.

Each MetadataSection element contains either an actual dialect-specific metadata element (line 6), for example wsd:definitions, or a pointer to another location, either on the Web (wsx:Location, line 8) or in a dedicated Metadata Web service (wsx:MetadataReference, lines 10-12).

The element wsx:Location, when present, contains a URI on which the piece of metadata with the specified dialect and identifier can be obtained. This allows a service to point to its metadata available on the Web.

The element wsx:MetadataReference, when present, contains a WS Addressing ([WSAddressing](#)) endpoint reference which identifies a service which will return the metadata upon a simple SOAP request.

The following example demonstrates metadata received from a stock-quote service.

```

01 <wsx:Metadata>
02   <wsx:MetadataSection
03     Dialect='http://schemas.xmlsoap.org/wSDL/'
04     Identifier='http://server.example.org/stockquote'>
05     <wsdl:definitions
06       name='StockQuote'
07       targetNamespace='http://server.example.org/stockquote'
08       xmlns:tns='http://server.example.org/stockquote'
09       xmlns:wsdl='http://schemas.xmlsoap.org/wSDL/'
10       xmlns:wsoap='http://schemas.xmlsoap.org/wSDL/soap' >
11       <wsdl:import namespace='http://server.example.org/stockquote'
12         location='http://server.example.org/stockquote' />
13       <wsdl:portType name='StockQuotePortType'>
14         <wsdl:operation name='GetLastTradePrice'>
15           <wsdl:input message='tns:GetLastTradePriceInput' />
16           <wsdl:output message='tns:GetLastTradePriceOutput' />
17         </wsdl:operation>
18       </wsdl:portType>
19       <wsdl:service name='StockQuoteService'>
20         <wsdl:port name='StockQuotePort'
21           binding='tns:StockQuoteBinding' >
22           <wsoap:address
23             location='http://server.example.org/endpoint' />
24         </wsdl:port>
25       </wsdl:service>
26     </wsdl:definitions>
27   </wsx:MetadataSection>
28
29   <wsx:MetadataSection
30     Dialect='http://schemas.xmlsoap.org/ws/2004/09/policy' >
31     <wsx:MetadataReference>
32       <wsa:Address>
33         http://server.example.com/policyOfTheDay/stockQuote
34       </wsa:Address>
35     </wsx:MetadataReference>
36   </wsx:MetadataSection>
37
38   <wsx:MetadataSection
39     Dialect='http://www.w3.org/2001/XMLSchema'
40     Identifier='http://server.example.com/schemas/stockQuote'>
41     <wsx:Location>
42       http://server.example.com/schemas/stockQuote
43     </wsx:MetadataReference>
44   </wsx:MetadataSection>
45 </wsx:Metadata>

```

Lines 5 to 26 contain the WSDL description of the service, wrapped in a metadata section with the WSDL dialect (<http://schemas.xmlsoap.org/wSDL/>) and the identifier being the target namespace of this specific WSDL (<http://server.example.org/stockquote>).

Lines 29 to 36 contain another metadata section pointing to a policy (dialect <http://schemas.xmlsoap.org/ws/2004/09/policy>) available by accessing the Web service at

<http://server.example.com/policyOfTheDay/stockQuote>. Finally lines 38 to 44 contain a metadata section pointing to a schema (dialect <http://www.w3.org/2001/XMLSchema>) with the target namespace <http://server.example.com/schemas/stockQuote> which is available on the Web at the same URI (compare lines 40 and 42).

3.2 Using WS-MEX to retrieve WSMO descriptions

In order for WS Metadata Exchange to be able to carry WSMO descriptions, WSMO must simply provide a dialect URI identifying this language and optionally also a mechanism for assigning identifier URIs to different WSMO descriptions. In a way similar to WSDL, the dialect URI can be the namespace of WSML/XML [de Bruijn 2004] elements, <http://www.wsmo.org/wsml/wsml-syntax#> but since WSML/XML files don't have any single identifying property (in particular, they don't have any targetNamespace notion), we cannot provide any value to the Identifier.

The following is a simple example of a WSML document in a WS-MEX metadata section:

```
01 <wsx:Metadata>
02   <wsx:MetadataSection
03     Dialect='http://wsmo.org/2004/wsml' >
04     <wsml:wsml xmlns:wsml="http://www.wsmo.org/wsml/wsml-syntax#" >
05       <wsml:ontology
06         name="http://example.org/ontologies/ticketing/">
07         <wsml:concept
08           name="Ticket"/>
09       </wsml:ontology>
10     </wsml:wsml>
11   </wsx:MetadataSection>
12 </wsx:Metadata>
```

4. Conclusions

This is an ongoing work and may be updated or split at any time. When it is split, this document will point to where the split parts are now, so this document should serve as a repository of specifications tying WSMO and Web services specifications.

To align WSMO with the WS specifications, this document introduces one new XML element:

wsml:serviceReference

used in WS Addressing endpoint references to point to a WSMO description of the service, also used in WS Policy to express a policy assertion assigning a referenced WSMO service description to the policy subject,

References

[de Bruijn 2004] J. de Bruijn (editor): *WSML/XML - An XML Syntax for WSML*, available at <http://www.wsmo.org/2004/d16/d16.3/v0.1/>

[Roman et al. 2004] D. Roman, U. Keller, H. Lausen (editors): *Web Service Modeling Ontology - Standard (WSMO - Standard)*, version 1.0 available at <http://www.wsmo.org/2004/d2/v1.0/>

[WSAddressing] D. Box, F. Curbera (editors) (2004): *Web Service Addressing*, W3C member submission, available at <http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>

[WSDL] R. Chinnici, M. Gudgin, J-J. Moreau, J. Schlimmer, S. Weerawarana (editors): *Web Services Description Language Part 1: Core Language*, W3C Last Call Working Draft available at <http://www.w3.org/TR/wsdl20/>

[WSMEX] F. Curbera, J. Schlimmer (editors) (2004): *Web Service Metadata Exchange*, available at <http://msdn.microsoft.com/ws/2004/09/mex/>

[WSPolicy] J. Schlimmer (editor): *Web Services Policy Framework*, available at <http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-policy.asp>

[XMLSchema] H. Thompson, D. Beech, M. Maloney, N. Mendelsohn (editors): *XML Schema part 1: Structures*, W3C Recommendation, 2001, available at <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

Acknowledgement

The work is funded by the European Commission under the projects ASG, DIP, EASAIER, enIRaF, Knowledge Web, Musing, Salero, Seemp, SemanticGOV, Super, SWING and TripCom; by Science Foundation Ireland under the D ERI-Lion Grant No.SFI/02/CE1/I13 ; by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the projects Grisino, RW², SemBiz, SeNSE 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.

Change Log

Date	Author	Description
2007/2/15	jacek	splitting policy out into d24.3, cleaning up the rest
2007/1/12	jacek	put in policy content created for DIP
2005/2/9	jacek	Decapitalized Web services, removed WSMX mention, hidden the conclusions, other editorial touches
2005/1/17	jacek	Reorganized section on Addressing, added ontology for endpoint references
2005/1/17	jacek	dropped sections on WSDL and XML Schema, pointed to a future grounding document instead
2005/1/17	jacek	added examples in section on Policy
2005/1/14	jacek	changed reference style for specifications Rationale: names of authors or editors of specifications are irrelevant, what is relevant is name of specification (which is unique and descriptive) or names of author companies. I don't think anybody will want to mention company names as reference.
2005/1/14	jacek	Added changelog



\$Date: 2007/02/15 15:36:50 \$

webmaster