WSDL-S: Specification, Tools, Use Cases and Applications

Amit P. Sheth  
Wright State University - Main Campus, amit@sc.edu

Kunal Verma  
Wright State University - Main Campus

Karthik Gomadam  
Wright State University - Main Campus

Follow this and additional works at: https://corescholar.libraries.wright.edu/knoesis

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation
https://corescholar.libraries.wright.edu/knoesis/663

This Presentation is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
WSDL-S: Specification, Tools, Use Cases and Applications
Amit Sheth, Kunal Verma, & Karthik Gomadam, University of Georgia

Introduction
• Plethora of standards and specifications to realize
  – Reuse
  – Interoperability
  – Composition

The SOA

Big Bang

Question

Can syntactic based standards realize the goals?

• The universe is not
• Heterogeneity

Mere XML based standardization is not sufficient

• Addressing is critical to interoperability

2006 Semantic Technology Conference
San Jose, California ● March 6-9, 2006
Process lifecycle semantics can help with some degree of automation in all the steps
1. Configuration and execution

namelessnerd, 2/1/2006
Look beyond syntax and XML

What does Semantics bring to the table?

- **Reuse**
  - Semantic descriptions of services to help find relevant services

- **Interoperability**
  - Beyond syntax to semantics (ontology based approach)

- **Composition**
  - Enable dynamic binding of partners

- **Some degree of automation across process lifecycle**
  - Process Configuration (Discovery and Constraint analysis)
  - Process Execution (Addressing run time heterogeneities like data heterogeneities.)
Juxtaposing the various approaches to SWS

<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
<th>Formalism</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWL-S</td>
<td>An OWL based upper ontology semantically representing Web services.</td>
<td>Description Logics</td>
</tr>
<tr>
<td>SWSL (Semantic Web Service Language)</td>
<td>A combination First Order Logic and rules to represent Web services.</td>
<td>First order logic, Different variants of rule languages (Horn, HiLog etc.)</td>
</tr>
<tr>
<td>WSDL-S</td>
<td>Use of extensibility elements in WSDL to annotate elements with terms in ontologies.</td>
<td>Agnostic (examples typically use description logics, but use of UML for conceptual modeling is also recognized)</td>
</tr>
<tr>
<td>WSMO</td>
<td>A F-Logic based conceptual model for representing Web services.</td>
<td>F-Logic</td>
</tr>
</tbody>
</table>

Each of these are W3C Member submissions

Semantics to Web Services: The ingredients

- **Conceptual Model**
  - An agreed upon model that captures the semantics of domain.
- **XML based service description**
  - Standards and specifications like WSDL for web service description, WS-Agreement for capturing agreements etc.
- **Annotate the service description**
What are the common ideas across them?
namelessnerd, 2/3/2006
WSDL-S: Specification, Tools, Use Cases and Applications
Amit Sheth, Kunal Verma, & Karthik Gomadam, University of Georgia

What is WSDL-S?

- WSDL: XML based service description
- Ontology capturing the semantics
- Annotate the WSDL with concepts from the ontology
supports higher degree of automation
namelessnerd, 2/1/2006

structural semantic interop.
namelessnerd, 2/1/2006
WSDL-S

- Offer an evolutionary and compatible upgrade of existing Web services standards

- Externalize the semantic domain models
  - agnostic to ontology representation languages.
  - reuse of existing domain models
  - allows annotation using multiple ontologies (same or different domain)

- updating tools around WSDL is relatively easier

Adding semantics to WSDL – guiding principles

- Build on existing Web Services standards

- Mechanism independent of the semantic representation language

- Mechanism should allow the association of multiple annotations written in different semantic representation languages
Guiding principles...

- Support semantic annotation of Web Services whose data types are described in XML schema with respect to concepts in ontologies.
- Provide support for rich mapping mechanisms between Web Service schema types and ontologies.

Why use WSDL-S

- Build on existing Web Services standards using only extensibility elements.
- Mechanism independent of the semantic representation language (though OWL is supported well).
- WSDL-S provides an elegant solution:
  - Help integration by providing mapping to agreed upon domain models (ontologies, standards like Rosetta Net, ebXML).
  - More detailed description by adding functional annotation.
- Ease in tool upgrades:
  - e.g., wsif / axis invocation.
add the four key aspects from CACM paper.
namelessnerd, 1/31/2006

namelessnerd, 1/31/2006
## Semantic annotations on WSDL elements

<table>
<thead>
<tr>
<th>Extension Element / Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modelReference (Element: Input and Output Message types)</td>
<td>Semantic annotation of WSDL input and output message types with concepts in a semantic model.</td>
</tr>
<tr>
<td>schemaMapping (Element: Input and Output Message types)</td>
<td>Association of structural and syntactic mappings between WSDL message types and concepts in a semantic model.</td>
</tr>
<tr>
<td>modelReference (Element: Operation)</td>
<td>Captures the semantics of the functional capabilities of an operation.</td>
</tr>
<tr>
<td>pre-conditions (Parent Element: Operation)</td>
<td>Set of semantic statements (or expressions represented using the concepts in a semantic model) that are required to be true before an operation can be successfully invoked.</td>
</tr>
<tr>
<td>effects (Parent Element: Operation)</td>
<td>Set of semantic statements (or expressions represented using the concepts in a semantic model) that must be true after an operation completes execution.</td>
</tr>
<tr>
<td>category (Parent Element: Operation)</td>
<td>Service categorization information that could be used when publishing a service in a Web Services registry such as UDDI.</td>
</tr>
</tbody>
</table>
PurchaseOrder.wsdl

......

<x:s:element name="processPurchaseOrderResponse" type="xs:string

wssem:modelReference="POOntology#OrderConfirmation"/>

</xs:schema>

</types>

<interface name="PurchaseOrder">

<wssem:category name="Electronics" taxonomyURI=http://www.naics.com/
taxonomyCode="443112" />

<operation name="processPurchaseOrder" pattern=wsdl:in-out

modelReference = "rosetta:#RequestQuote">

<input messageLabel = "processPurchaseOrderRequest"

element="tns:processPurchaseOrderRequest"/>

<output messageLabel = "processPurchaseOrderResponse"

element="processPurchaseOrderResponse"/>

</operation>

</interface>

Annotating operations

- extension element : Precondition
  - A set of assertions that must be satisfied before a Web service operation can be invoked
    - "must have an existing account with this company"
    - "only US customers can be served"

- extension element : Effect
  - Defines the state of the world/information model after invoking an operation.
    - "item shipped to mailing address"
    - "the credit card account will be debited"

- extension attribute : Category
  - Models a service category on a WSDL interface element.
    - category = "Electronics" Code = "naics:443112"

- extension element : Action
  - Annotated with a functional ontology concept.
    - action = "Rosetta:RequestQuote"
Annotating message types - simple correspondences

1:1 Correspondences

```xml
<wsdl:types>
  <xs:element name="processPurchaseOrderResponse" type="xs:string"/>
</wsdl:types>
```

OWL ontology

1:1 Correspondences

```xml
<xs:element name="processPurchaseOrderResponse" type="xs:string" wssem:modelReference="POOntology#OrderConfirmation"/>
```

Annotating message types - complex correspondences

1. modelReference to establish a semantic association
2. schemaMapping to resolve structural heterogeneities beyond a semantic match
Using modelReference and schemaMapping

- **modelReference at the complex type level**
  - Typically used when specifying complex associations at leaf level is not possible
  - Allows for specification of a mapping function

- **modelReference at the leaf levels**
  - Assumes a 1:1 correspondence between leaf elements and domain model concepts
Representing mappings

```xml
<complexType name="POAddress" wssem:schemaMapping="http://www.ibm.com/schemaMapping/POAddress.xsl#input-doc=doc("POAddress.xml")">
  <all>
    <element name="streetAddr1" type="string" />
    <element name="streetAdd2" type="string" />
    <element name="poBox" type="string" />
    <element name="city" type="string" />
    <element name="zipCode" type="string" />
    <element name="state" type="string" />
    <element name="country" type="string" />
    <element name="recipientInstName" type="string" />
  </all>
</complexType>
```

Mapping using XSLT

```xml
<xsl:template match="/">
  <POOntology:Address rdf:ID="Address1">
    <POOntology:has_StreetAddress rdf:datatype="xs:string">
      <xsl:value-of select="concat(POAddress/streetAddr1, POAddress/streetAdd2)"/>
    </POOntology:has_StreetAddress>
    <POOntology:has_City rdf:datatype="xs:string">
      <xsl:value-of select="POAddress/city"/>
    </POOntology:has_City>
    <POOntology:has_State rdf:datatype="xs:string">
      <xsl:value-of select="POAddress/state"/>
    </POOntology:has_State>
  </POOntology:Address>
</xsl:template>
```

WSDL-S evolution

- **Extension Adaptation**
- **Action Attribute for Functional Annotation**
- **Can use XML, OWL or UML types**
- **Pre and Post Conditions**
I6 I recommend using XSLT, which is a bit more readable and less overwhelming than XQuery functions.

</Rama>
IBM_USER, 5/27/2005

Slide 24

I1 In the interest of time, this could be dropped, but we need to be sure to mention that the modelReferences are independent of the ontology representation language.

IBM_USER, 5/27/2005
Using WSDL-S in Web Process Lifecycle

WSDL-S in the life cycle of a Web service
WSDL-S in the life cycle of a Web process

WSDL-S for advanced lifecycle support

- Data Mediation support
  - with Axis2 implementation
- Dynamic Process Binding
  - Process template
- Adaptation
Interoperability in Web services

- Impediments beyond semantic composition of Web services
  - Message level heterogeneities between communicating Web services

Message level Heterogeneities

- Syntactic - differences in the language used for representing the elements
- Model/Representational - differences in the underlying models (database, ontologies) or their representations (relational, object-oriented, RDF, OWL)
- Structural - differences in the types, structures of the elements
- Semantic - where the same real world entity is represented using different terms (or structures) or vice versa

Resolved by the XML based environment

WSDL-S: Semi-automatic solution
Abstract Process Specification for Dynamic Configuration

1. Specify process control flow by using virtual partners.
2. Capture Functional Requirements of Services using Semantic Templates.

- Semantic Templates capture the functionality of a Web service with the specific ontologies/other domain models.
- Find a service that is located in Athens, GA. It must allow the user to return and cancel if needed.
- The template can capture both functional and non-functional (QoS) requirements such as response time, security, etc.

Matching
Mapping
A lot of early work on heterogeneous database integration is still quite useful.
Semantic Templates

- Semantic Templates capture the functionality of a Web service with the help of ontologies/other domain models.
- Find a service that sells RAM in Athens, GA. It must allow the user to return and cancel, if needed.
- The template can also have non-functional (QoS) requirements such as response time, security, etc.

Sample Semantic Template

Service Level Metadata
IndustryCategory = NAICS: Electronics
ProductCategory = DUNS: RAM
Location = Athens, GA

Semantically Defined Operations

Operation1 = Rosetta#requestPurchaseOrder
  Input = Rosetta#PurchaseOrderDetails
  Output = Rosetta#PurchaseConfirmation
  ResponseTime < 5s

Operation2 = Rosetta#CancelOrder

Operation3 = Rosetta#ReturnProduct

WSDL-S is used to capture semantic templates.

Realizing Dynamic Configuration

Discovery Results

After ILP

Candidate Services with Constraints

- RAM Candidate Service 1 (R1)
  - Cost = $200
  - Supply Time = 3 Days
- RAM Candidate Service 2 (R2)
  - Cost = $200
  - Supply Time = 7 Days
- HD Candidate Service 1 (H1)
  - Cost = $500
  - Supply Time = 7 Days
- HD Candidate Service 2 (H2)
  - Cost = $200
  - Supply Time = 15 Days
- HD Candidate Service 3 (H3)
  - Cost = $500
  - Supply Time = 15 Days

Compatible Service Sets in Increasing Cost Order

1. HD, HD
   Cost = $1500
   Process Constraints:
   - Supply Time < 7 Days
   - L: Prefer(HD1) = True
   - Min. Cost

2. HD, R1
   Cost = $300
   Process Constraints:
   - Supply Time < 3 Days
   - L: Prefer(R1) = True
   - Min. Cost

3. HD, R2
   Cost = $1700
   Process Constraints:
   - Supply Time < 7 Days
   - L: Prefer(R2) = True
   - Min. Cost

After SWRL

ILP Solver

Constraint Analyzer

SWRL Reasoner

Process Constraints

- Supply Time < 7 Days
- L: Prefer(HD1) = True
- Min. Cost

After SWRL
International Bank Use Case

• This bank is considering moving to SOA based architecture

They feel WSDL has following shortcomings
  - Schema level
    • Unable to define well known restrictions: email, credit card number
    • Unable to define detail description for enumerations: SPD for Summary Plan Description

• WSDL operation level
  • pre-conditions and post-conditions of a service operation
  • restrictions on elements / complexTypes that are operation specific (e.g. customerId in CustomerType must be null for AddCustomer; but it's mandatory for GetCustomer)
Use Case Details

- A search service is defined to search by either personal name or commercial name.
- The search engine would return at least one element of names, or a SOAP fault.

Adding Contracts to WSDL

- In the use case, it’s expressed as the following:
  - A name has to be provided.
  - If it’s a personal name, either last name or personal name must exist.
  - If it’s a commercial name, either corporate name or stock ticker must exist.
  - Either at least one or no more than 100 names would be returned, or an error “not found” will occur.

Currently, we are working with the bank to realize this case using WSDL-S.
International Bank Use Case

- This bank is considering moving to SOA based architecture

- They feel WSDL has following shortcomings
  - Schema level
    - Unable to define well known restrictions: email, credit card number
    - Unable to define detail description for enumerations: SPD for Summary Plan Description
  - WSDL operation level
    - pre-conditions and post-conditions of a service operation
    - restrictions on elements / complexTypes that are operation specific (e.g. customerId in CustomerType must be null for AddCustomer; but it’s mandatory for GetCustomer)

Use Case Details

- A search service is defined to search by either personal name or commercial name.
- The search engine would return at least one element of names, or a SOAP fault
Adding Contracts to WSDL

- In the use case, it’s expressed as the following:
  - A name has to be provided.
  - If it’s a personal name, either last name or personal name must exist.
  - If it’s a commercial name, either corporate name or stock ticker must exist.
  - Either at least one or no more than 100 names would be returned, or an error “not found” will occur.

These can be represented as preconditions in WSDL-S

Current Agri-Marketing in India

- Seller: Farmer
- Agriculture Produce Market Committee
- Brokers associated With APMC
- Buyers
Current Agri-Marketing scenario in India

- A farmer can sell his produce to either Agriculture Produce Market Committees or Brokers associated with APMC’s.
- APMC’s sell the produce either by retail or in open markets.
- Research is underway in creating SOA based architectures to realize the buyer seller interactions as services.

Current Agri-Marketing scenario in India

- Farmers use kiosks to interact with the buyer services
- Farmers need to locate the right APMC for their products
  - Some APMC’s may not have refrigeration making them unsuitable for fresh vegetables, diary products etc.
  - Farmers might want to get paid in cash the same day whilst some APMC’s may not be willing to do so.
- Farmers use the web based interface to then sell their produce to the APMC.
**Why WSDL-S?**

- Uses semantics to provide richer descriptions of the services offered by APMC’s.
  - An APMC buys wheat, potatoes, fresh meat and dairy products. The APMC can use WSDL-S to represent this information in his service description.
  - Allows for capturing policies such as "Refrigeration is free for 2 business days" or "Same day payment will be issued in cash"
- Various APMC’s have varying data definitions. It is hard to create a client that can interoperate, due to heterogeneities that are present. WSDL-S help address them by mediation.

**Using WSDL-S in Bioinformatics**

- **ProPreO - Experimental Proteomics Process Ontology (CCRC / LSDIS)**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:ngp"
xmlns:sem="http://www.ibm.com/xmlns/WebServices/WSSemantics"
xmlns:ProPreO="http://lsdis.cs.uga.edu/ontologies/ProPreO.owl">
  <wsdl:types>
    <schema targetNamespace="urn:ngp"
      xmlns="http://www.w3.org/2001/XMLSchema">
      ......
    </schema>
  </wsdl:types>
  <wsdl:message name="replaceCharacterRequest" wssem:modelReference="ProPreO#peptide_sequence">
    <wsdl:part name="in0" type="soapenc:string"/>
    <wsdl:part name="in1" type="soapenc:string"/>
    <wsdl:part name="in2" type="soapenc:string"/>
  </wsdl:message>
</wsdl:definitions>
```

Excerpt: Bio-Informatics Web service WSDLS

Excerpt: ProPreO – process ontology
Making them work: Tooling

METEOR-S Tools

- Illumina
  - Semantic Web Service Discovery
- Radiant
  - Semantic Annotation and publication of Web Services
- SAROS
  - Semantic Template based Process Designer
- IBM Alphaworks
  - Eclipse plug-ins for semantic matching and composition of WSDL-S based Web services as part of Emerging Technologies Toolkit (ETTK)
WSDL-S collaborations

- **METEOR-S collaboration with WSMO**
  - Using WSDL-S for grounding Web services annotated with WSML ontologies

- Influencing OASIS / W3C

W3C Charters Proposed

- Charter of the Semantic Annotations for WSDL Working Group
  - Primary objective to use WSDL’s extensibility mechanism to add more information to data definitions in WSDL
  - Also, recognizes WSDL-S as an important input
<Joel>Points I would make:

If more revolutionary approaches are pursued, we must be sure that existing WSDL, XML Schemas for business documents, and the tools that exploit them can still be leveraged.

WSDL-S could be that bridge.

This is a key success factor.

IBM_USER, 5/27/2005
Conclusions and The Next Steps

- Simplicity is the always the key for adoption
- WSDL-S may be the evolutionary solution for some real problems – reuse and interoperability
- Although preliminary work and prototyping is completed, more work needs to be done
  - Use cases
  - Tooling support
  - Greater adoption by vendors

More information

- WSDL-S resource including tools and use cases: http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/