5-10-2007

Semantic Annotations for WSDL

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

Jacek Kopecky

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

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 corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
SAWSDL: Tools and Applications

Amit P. Sheth
Kno.e.sis Center
Wright State University, Dayton, OH
Knoesis.wright.edu

Key contributors: Karthik Gomadam, Kunal Verma, Ajith Ranabahu, Meena Nagarajan
What does Semantics bring to the table?

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

- **Better Interoperability**
  - Beyond syntax to semantics, mapping of data exchanged between the services (very time consuming without semantics, just as XML in WSDL gives syntactic interoperability, SAWSDL gives semantic interoperability)

- **Configuration/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 and exceptions)
What can we support or demonstrate today

- API for handling SAWSDL documents: [SAWSDL4J](#)
- Tool for annotating WSDL services to produce SAWSDL: [Radiant](#) and for discovery: Lumina
- Using SAWSDL with UDDI for Discovery: [SemBowser](#)
- Using SAWSDL with Apache Axis for Data Mediation
- Using SAWSDL with WS-BPEL for run-time binding
- Early Examples of SAWSDL annotated services: biomedical research

Also:

- [Semantic Tools for Web Services](#) by IBM alphaWorks
- [WSMO Studio](#), more mentioned by Jacek
Semantic Web Services Discovery: Illumia

UDDI - UDDI editor - Eclipse SDK

Operations
Input Operations
- Finding operations
- operation1
  - semantic input
  - semantic output
  - precondition
  - effect

discovered services
- discover result
  - service: CurrencyConvert
  - service: CurrencyExchangeService

Semantic templates
- partner services
  - BaseRealTimeQuotes
  - CurrencyConverter
  - CurrencyExchangeService
  - CurrencyRates
  - DelayedStockQuote
  - DTSFastQuote
  - NexusStudio_x0020_Stock_x0020_Quote
  - StockQuote
  - StockQuotes
  - StockServices
Syntactic and Semantic Match do not suffice

DATA MEDIATION REQUIRED

Address line 1
Address line 2
City_State_zip

Listing Name
First Name
Last Name
Address
City
State
Postal Code
Phone Number
Published

INPUT TO WEB SERVICE 2

OUTPUT FROM WEB SERVICE 1

Web service 1
Address Lookup

OUTPUT FROM WEB SERVICE 2

Census Track
State Number
County Number
Block Number
Block Group

INPUT TO WEB SERVICE 1

Telephone Number
Mediation approach

- User specified mappings from Web service message element to semantic model concept (say OWL Ontology)
  - upcast: from WS message element to OWL concept
  - Downcast: from OWL concept to WS message element

```xml
<complexType name="Address">
  <sequence>
    <element name="StreetAd1" type="xsd:string"/>
    <element name="StreetAd2" type="xsd:string"/>
    ...........
  </sequence>
</complexType>
```

```xml
<POOntology:has_StreetAddress rdf:datatype="xs:string">{ fn:concat($a/streetAddr1 , " ", $a/streetAddr2 ) }</POOntology:has_StreetAddress>
```
### Domain Incompatibilities

#### Naming conflicts
- Two attributes that are semantically alike might have different names (synonyms)
- Two attributes that are semantically unrelated might have the same names (homonyms)

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student(Id#, Name)</td>
<td>Student(SSN, Name)</td>
</tr>
<tr>
<td>Student(Id#, Name)</td>
<td>Book (Id#, Name)</td>
</tr>
</tbody>
</table>

A semantic annotation on the entities and attributes (provided by WSDL-S:modelReference) will indicate their semantic similarities.

#### Data representation conflicts
- Two attributes that are semantically similar might have different data types or representations

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student(Id#, Name)</td>
<td>Student(Id#, Name)</td>
</tr>
<tr>
<td>Marks 1-100</td>
<td>Grades A-F</td>
</tr>
</tbody>
</table>

* Mapping WS2 Id# to WS1 Id# is easy with some additional context information while mapping in the reverse direction is most likely not possible.

#### Data scaling conflicts
- Two attributes that are semantically similar might be represented using different precisions

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICKET (FlightNo, MovieName)</td>
<td>TICKET (FlightNo, Arr. Airport, Dep. Airport)</td>
</tr>
<tr>
<td>EMPLOYEE (Id#, Name)</td>
<td>WORKER (Id#, Name)</td>
</tr>
</tbody>
</table>

A semantic annotation on the entities and attributes (provided by WSDL-S:modelReference) will indicate their semantic similarities.

### Entity Definition

#### Naming conflicts
- Semantically alike entities might have different names (synonyms)
- Semantically unrelated entities might have the same names (homonyms)

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICKET (TicketNo, MovieName)</td>
<td>TICKET(FlightNo, Arr. Airport, Dep. Airport)</td>
</tr>
<tr>
<td>PERSON (Name, Address, HomePhone, WorkPhone)</td>
<td>PERSON (Name, Address, Phone)</td>
</tr>
</tbody>
</table>

A semantic annotation on the entities and attributes (provided by WSDL-S:modelReference) will indicate their semantic similarities.

#### Schema Isomorphism conflicts
- Semantically similar entities may have different number of attributes

* Mapping in both directions will require some additional context information.

### Abstraction Level Incompatibility

#### Generalization conflicts
- Semantically similar entities are represented at different levels of generalization in two Web services

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAD-STUDENT (ID, Name, Major)</td>
<td>STUDENT (ID, Name, Major, Type)</td>
</tr>
</tbody>
</table>

* WS2 defines the student entity at a much general level. A mapping from WS1 to WS2 requires adding a Type element with a default ‘Graduate’ value, while mapping in the other direction is a partial function.

#### Aggregation conflicts
- Semantically similar entities are represented at different levels of generalization in two Web services

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROFESSOR (ID, Name, Dept)</td>
<td>FACULTY (ID, ProfID, Dept)</td>
</tr>
</tbody>
</table>

* A set-of Professor entities is a Faculty entity. When the output of WS1 is a Professor entity, it is possible to identify the Faculty group it belongs to, but generating a mapping in the other direction is not possible.

#### Attribute Entity conflicts
- Semantically similar entity modeled as an attribute in one service and as an entity in the other

<table>
<thead>
<tr>
<th>Web service 1</th>
<th>Web service 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>COURSE (ID, Name, Semester)</td>
<td>DEPT (Course, Sem, ...)</td>
</tr>
</tbody>
</table>

* Course modeled as an entity by WS1 is modeled as an attribute by WS2. With definition contexts, mappings can be specified in both directions.

* Interoperation between services needs transformation rules (mapping) in addition to annotation of the entities and/or attributes indicating their semantic similarity (matching).
Mediation approach continued...

- Web services interoperate by re-using these mappings.
  - Ontologies now a vehicle for Web services to resolve message level heterogeneities
DM Architecture components

• **METEOR-S Middleware**
  – **EPR handler – End Point Resolution handler**
    • For clients to use the middleware
    • Reroute SOAP messages to middleware

  – **DM handler – Data Mediation handler**
    • Main component for facilitating data mediation
    • Works with the EPR handler + a mapping processing engine
      (SAXON for XQuery / XSLT)

• Uses extensibility support offered by Axis 2 (handlers)
Semantic Templates

- **SAWSDL + Enhanced policy descriptions** to model the data, functional and non-functional semantics at the various tiers
  - Business Process Tier: Capture process level requirements
  - Implementation Tier: Capture partner level requirements
- Non-functional semantics captured at template and operation levels.
- **XML representation** for interoperability.
Semantic Templates

- SAWSDL for data and functional semantics
- Semantic Policy Descriptions for non-functional semantics
Example of a semantic template in the supply chain domain
Semantic Discovery

- Finds actual services matching semantic templates
- Implemented as a layer over UDDI
- Current implementation based on ontological representation of operations, inputs and outputs.
- Returns ranked of services for each semantic template
USING SAWSDL WITH WS-BPEL FOR RUN-TIME BINDING
Dynamic configuration Problem
Find optimal partners for the process based on process constraints – cost, supply time, etc.

Conceptual Approach
1. Create framework to capture represent domain knowledge
2. Represent constraints on the domain knowledge
3. Ability to reason on the constraints and configure the process
Dynamic Binding: Guiding principles

- Semantic templates to capture the requirements for each partner.
- Partners are selected during the run time of the process and the process is configured
  - Semantically Enhanced UDDI Registries for discovery of partners.
  - Approaches to match enhanced policies (Sem-Pol) and agreements (SWAPS)
- Execution environment supporting discovery, configuration and invocation.
Example of a process with semantic templates
Semantic Biological Web Services Registry

Web services discovery using task name

This allows the user to search for Web services with given ‘Task Name’. This requires an exact match between the user defined term and task concept of the service.

Task Name: RAW to mzXML

Submit | Reset
Semantic Web Services

- Formalize description and classification of Web Services using ProPreO concepts

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:ngp"

xmlns:wssem="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">

<complexType>

</complexType>
</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>
```
ISiS – Integrated Semantic Information and Knowledge System

Semantic Web Process to incorporate provenance

1. Biological Sample Analysis by MS/MS
2. Raw Data to Standard Format
3. Data Pre-process
4. DB Search (Mascot/Sequest)
5. Results Post-process (ProValt)

Semantic Annotation Applications

- Raw Data
- Standard Format Data
- Filtered Data
- Search Results
- Final Output

Storage

Biological Information

Knowledge Enabled Information and Services Science
• **Evaluate the specific effects of changing a biological parameter:** Retrieve *abundance* data for a given *protein* expressed by three different *cell types* of a specific *organism*.

• **Retrieve raw data supporting a structural assignment:** Find all the raw *ms data files* that contain the *spectrum* of a given *peptide sequence* having a specific *modification* and *charge state*.

• **Detect errors:** Find and compare all *peptide* lists identified in *Mascot output files* obtained using a similar *organism*, *cell-type*, *sample preparation protocol*, and *mass spectrometry conditions*.

**ProPreO concepts highlighted in red**
Some Relevant Papers


Stargate Portal: SemBowser and example SAWSDL service: http://glycomics.ccrc.uga.edu/stargate/index.jsp