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
Syntactic and Semantic Match do not suffice
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>

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

### Domain Incompatibilities – attribute level differences that arise because of using different descriptions for semantically similar attributes

<table>
<thead>
<tr>
<th>Heterogeneities / Conflicts</th>
<th>Examples - conflicted elements shown in color</th>
<th>Suggestions / Issues in Resolving Heterogeneities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Naming conflicts</strong></td>
<td>Web service 1 Student(id#, Name)</td>
<td>A semantic annotation on the entities and attributes (provided by WSDL-SModelReference) will indicate their semantic similarities.</td>
</tr>
<tr>
<td>Two attributes that are semantically alike might have different names (synonyms)</td>
<td>Web service 2 Student(SSN, Name)</td>
<td></td>
</tr>
<tr>
<td>Two attributes that are semantically unrelated might have the same names (homonyms)</td>
<td>Web service 1 Student(id#, Name)</td>
<td></td>
</tr>
<tr>
<td>Web service 1 Student(id#, Name)</td>
<td>Web service 2 Student(id#, Name)</td>
<td>* Mapping WS2 id# to WS1 id# is easy with some additional context information while mapping in the reverse direction is most likely not possible.</td>
</tr>
<tr>
<td><strong>Data representation conflicts</strong></td>
<td>Web service 1 Student(id#, Name)</td>
<td></td>
</tr>
<tr>
<td>Two attributes that are semantically similar might have different data types or representations</td>
<td>Web service 2 Student(id#, Name)</td>
<td></td>
</tr>
<tr>
<td>Web service 1 Student(id#, Name)</td>
<td>Web service 2 Student(id#, Name)</td>
<td>* Mapping WS2 id# to WS1 id# is easy with some additional context information while mapping in the reverse direction is most likely not possible.</td>
</tr>
<tr>
<td>Id# defined as a 4 digit number</td>
<td>Id# defined as a 9 digit number</td>
<td></td>
</tr>
<tr>
<td><strong>Data scaling conflicts</strong></td>
<td>Web service 1 Marks 1-100</td>
<td></td>
</tr>
<tr>
<td>Two attributes that are semantically similar might be represented using different precisions</td>
<td>Web service 2 Grades A-F</td>
<td></td>
</tr>
<tr>
<td>Web service 1 Marks 1-100</td>
<td>Web service 2 Grades A-F</td>
<td></td>
</tr>
</tbody>
</table>

### Entity Definition – entity level differences that arise because of using different descriptions for semantically similar entities

<table>
<thead>
<tr>
<th>Heterogeneities / Conflicts</th>
<th>Examples - conflicted elements shown in color</th>
<th>Suggestions / Issues in Resolving Heterogeneities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Naming conflicts</strong></td>
<td>Web service 1 EMPLOYEE (id#, Name)</td>
<td>A semantic annotation on the entities and attributes (provided by WSDL-SModelReference) will indicate their semantic similarities.</td>
</tr>
<tr>
<td>Semantically alike entities might have different names (synonyms)</td>
<td>Web service 2 WORKER (id#, Name)</td>
<td></td>
</tr>
<tr>
<td>Semantically unrelated entities might have the same names (homonyms)</td>
<td>Web service 1 TICKET (FlightNo, MovieName)</td>
<td></td>
</tr>
<tr>
<td>Web service 1 TICKET (FlightNo, MovieName)</td>
<td>Web service 2 TICKET (FlightNo, Arr. Airport, Dep. Airport)</td>
<td></td>
</tr>
<tr>
<td><strong>Schema Isomorphism conflicts</strong></td>
<td>Web service 1 PERSON (Name, Address, HomePhone, WorkPhone)</td>
<td>* Mapping in both directions will require some additional context information.</td>
</tr>
<tr>
<td>Semantically similar entities may have different number of attributes</td>
<td>Web service 2 PERSON (Name, Address, Phone)</td>
<td></td>
</tr>
<tr>
<td>Web service 1 PERSON (Name, Address, HomePhone, WorkPhone)</td>
<td>Web service 2 PERSON (Name, Address, Phone)</td>
<td></td>
</tr>
</tbody>
</table>

### Abstraction Level Incompatibility – Entity and attribute level differences that arise because two semantically similar entities or attributes are represented at different levels of abstraction

<table>
<thead>
<tr>
<th>Heterogeneities / Conflicts</th>
<th>Examples - conflicted elements shown in color</th>
<th>Suggestions / Issues in Resolving Heterogeneities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generalization conflicts</strong></td>
<td>Web service 1 GRADSTUDENT (id, Name, Major)</td>
<td>* 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.</td>
</tr>
<tr>
<td>Semantically similar entities are represented at different levels of generalization in two Web services</td>
<td>Web service 2 STUDENT (id, Name, Major, Type)</td>
<td></td>
</tr>
<tr>
<td><strong>Aggregation conflicts</strong></td>
<td>Web service 1 PROFESSOR (id, Name, Dept)</td>
<td>* 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.</td>
</tr>
<tr>
<td>Semantically similar entities are represented at different levels of generalization in two Web services</td>
<td>Web service 2 FACULTY (id, ProfId, Dept)</td>
<td></td>
</tr>
<tr>
<td><strong>Attribute Entity conflicts</strong></td>
<td>Web service 1 COURSE (id, Name, Semester)</td>
<td>* Course modeled as an entity by WS1 is modeled as an attribute by WS2. With definition contexts, mappings can be specified in both directions.</td>
</tr>
<tr>
<td>Semantically similar entity modeled as an attribute in one service and as an entity in the other</td>
<td>Web service 2 DEPT (Course, Sem, …)</td>
<td></td>
</tr>
</tbody>
</table>

* 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 Template

ServiceLevelMetaData (SLM)
Category= NAICS:Electronics
ProductCategory= DUN3:RAM
Location= Athens_GA

SemanticOperation Template (SOTP1)
Action= Rosetta:RequestPurchaseOrder
Input= Rosetta:PurchaseOrder_Input
Output= Rosetta:PurchaseOrder_Output
OLP= (Encryption = RSA, ResponseTime< 5 Sec)

SemanticOperation Template (SOTP2)
Action= Rosetta:CancelOrder
Input= Rosetta:CancelOrder_Input
Output= Rosetta:CancelOrder_Output
OLP= (Encryption = RSA, ResponseTime< 5 Sec)
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

STARGATE

Welcome to SemBROWSER, 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 mXML

Submit  Reset

STARGATE

Welcome to SemBROWSER, Semantic Biological Web Services Registry

Name: SysJavaRawmzXMLService
WSDL Location: http://192.168.168.100:8080/axis2/Service/Commv0.21/SysJavaRawmzXML.jsr/glycomics
Business Entity: Glycomics
• 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>
  <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:types>
</wsdl:definitions>
```
ISiS – Integrated Semantic Information and Knowledge System

Semantic Web Process to incorporate provenance

Biological Sample Analysis by MS/MS → Raw Data to Standard Format → Data Pre-process → DB Search (Mascot/Sequest) → Results Post-process (ProValt)

Semantic Annotation Applications

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

Storage

Biological Information
• **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.

**A Web Service Must Be Invoked**

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


**Stargate Portal: SemBowser and example SAWSDL service:**

http://glycomics.ccrc.uga.edu/stargate/index.jsp