



LSDIS

Large Scale Distributed Information Systems



University of Georgia
Computer Science Department

Web Service Semantics-WSDL-S

Meenakshi Nagarajan

for the

[WSDL-S](#) team

R. Akkiraju*, J. Farrell*, J. Miller, M. Nagarajan, M. Schmidt*, A. Sheth, K. Verma
"Web Service Semantics - WSDL-S"

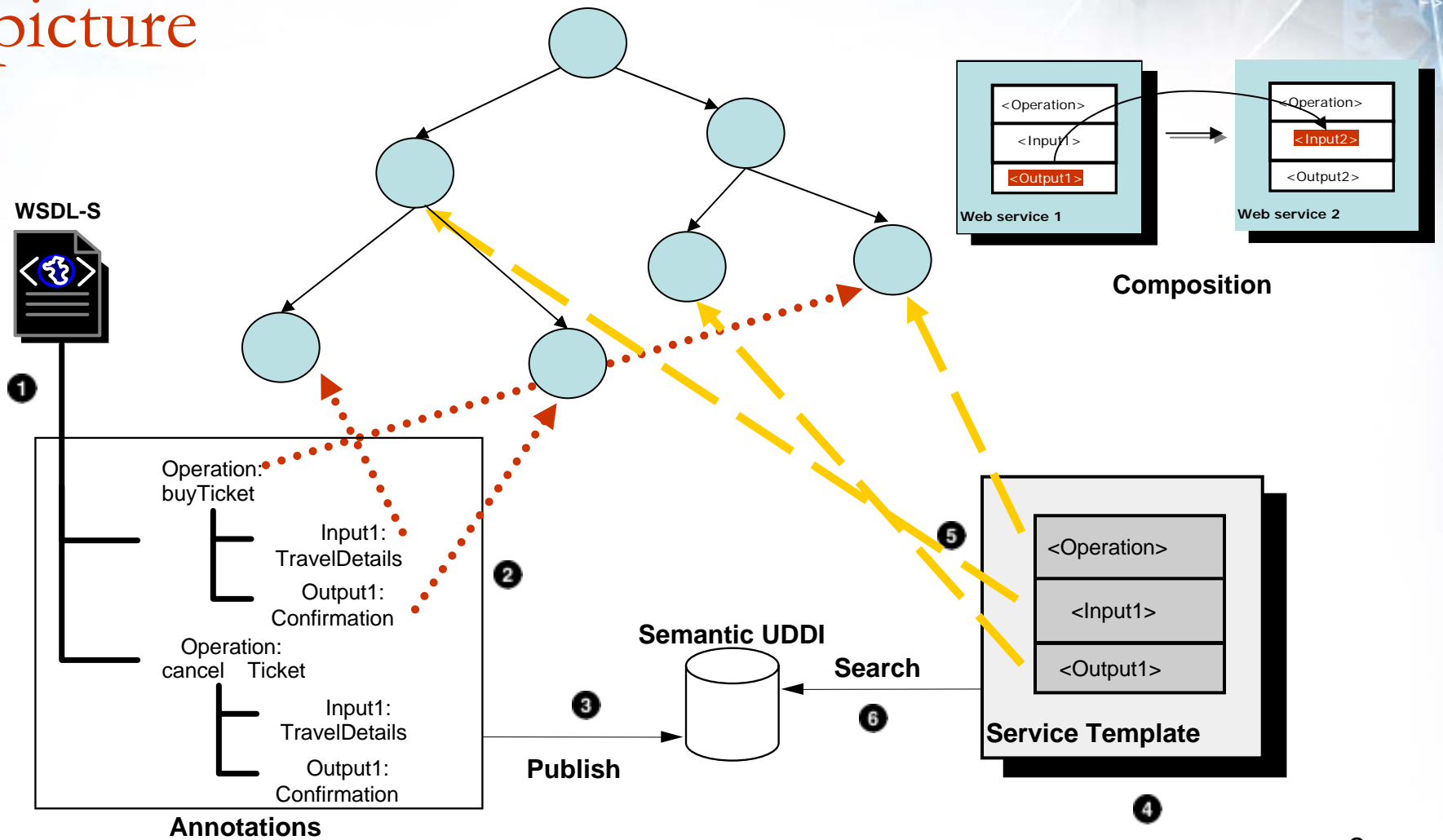
A joint UGA-IBM* Technical Note, version 1.0, April 18, 2005. <http://lsdis.cs.uga.edu/projects/METEOR-S/WSDL-S>

[http://www.alphaworks.ibm.com/g/g.nsf/img/semanticsdocs/\\$file/wssemantic_annotation.pdf](http://www.alphaworks.ibm.com/g/g.nsf/img/semanticsdocs/$file/wssemantic_annotation.pdf)



IBM is the most active research organization in the world. Very little is built upon research. TEORIS: Semantic Web Services

WSDL-S : scope, proposal and the bigger picture





IBM, the most active research
Very much build upon research
TECHNICAL SERVICES Web Services

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



IBM, the most active research
Very much build upon research
TECIS/S2 Semantics Web Service

Guiding principles...

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



Top of the most active research
Very high build upon research
TEOIS/S2 Semantics Web Services

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



Semantic annotations on WSDL elements

- Annotating message types (XSD complex types and elements)
 - extension attribute : modelReference (semantic association)
 - extension attribute : schemaMapping (schema/data mapping)
- Annotating operations
 - extension elements : precondition and effect (child elements of the operation element)
 - extension attribute : category (on the interface element)
 - extension attribute : modelreference (action) (on operation element)



PurchaseOrder.wsdl

```
.....  
<xs: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"/>  
  
<!--Precondition and effect are added as extensible elements on an operation-->  
<wssem:precondition name="ExistingAcctPrecond"  
wssem:modelReference="POOntology#AccountExists">  
<wssem:effect name="ItemReservedEffect"  
wssem:modelReference="POOntology#ItemReserved">  
</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”

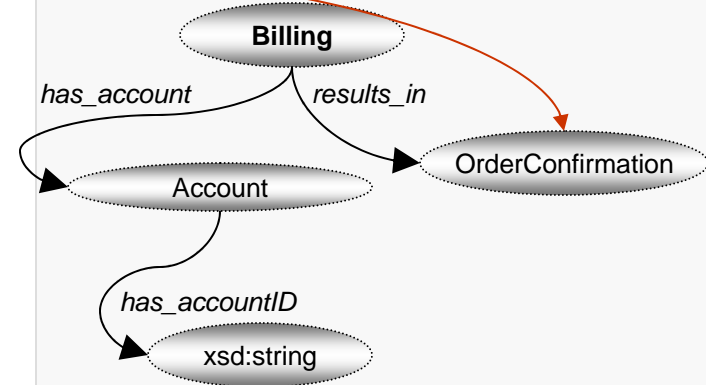


Annotating message types - simple correspondences

semantic match

```
<wsdl:types>  
  (...)  
  <xs:element name="processPurchaseOrderResponse"  
    type="xs:string"  
    (...)  
</wsdl:types>
```

WSDL message element



OWL ontology

1:1 Correspondences

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

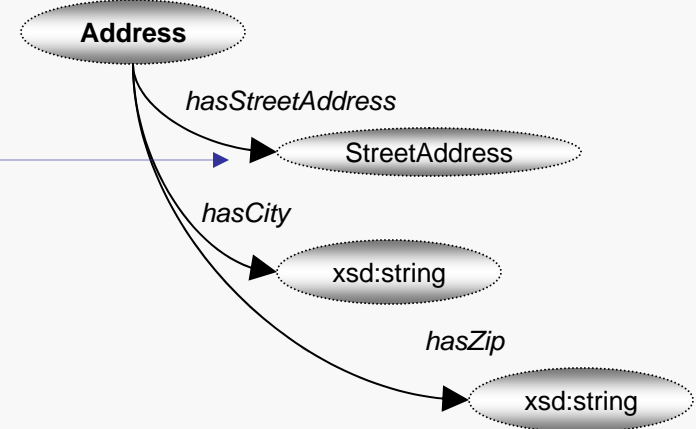


Annotating message types - complex correspondences

semantic match

```
<wsdl:types>
  (...)
  <complexType name="Address">
    <sequence>
      <element name="StreetAd1" type="xsd:string"/>
      <element name="StreetAd2" type="xsd:string"/>
      .....
    </sequence>
  </complexType>
  (...)
</wsdl:types>
```

WSDL complex type element



OWL ontology

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



Using modelReference and schemaMapping

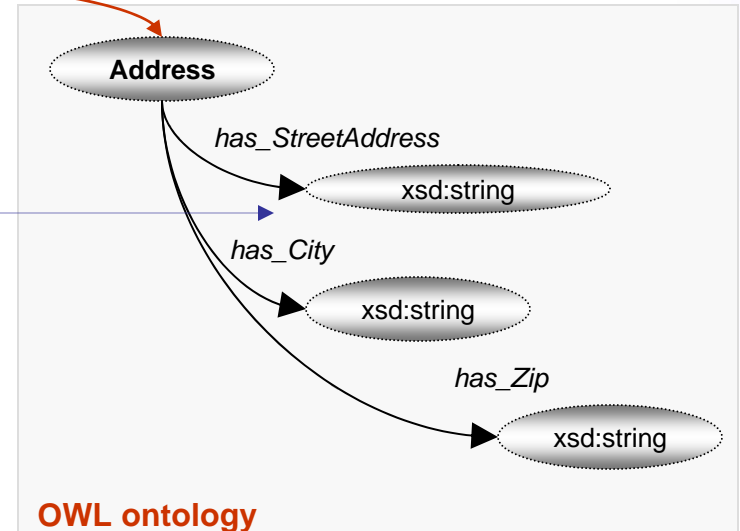
- **modelReference at the complex type level**

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

semantic match

```
<complexType name="POAddress"  
  wssem:modelReference="POOntology#Address"  
  wssem:schemaMapping="http://www.ibm.com/schemaMapping/POAdd  
  res.xq#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>
```

WSDL complex type element





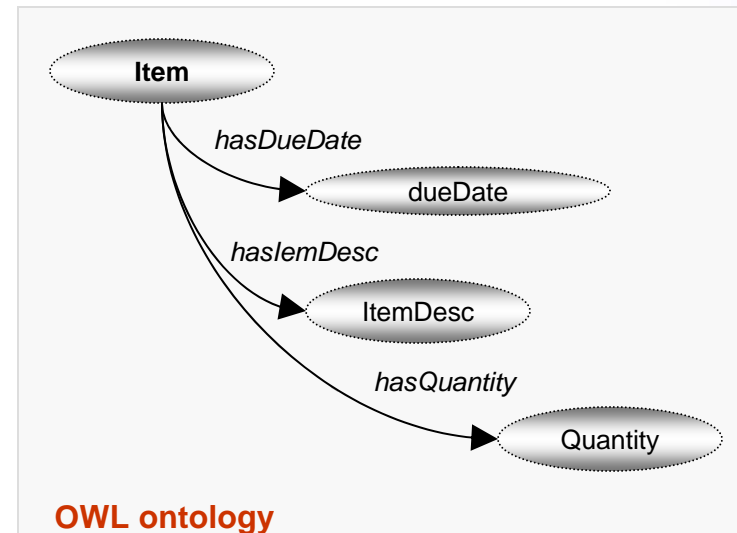
Using modelReference and schemaMapping

- **modelReference at the leaf levels**

- assumes a 1:1 correspondence between leaf elements and domain model concepts

```
<complexType name="POItem" >
<all>
  <element name="dueDate" nillable="true" type="dateTime"
    wssem:modelReference="POOntology#DueDate"/>
  <element name="qty" type="float"
    wssem:modelReference="#POOntology#Quantity"/>
  <element name="EANCode" nillable="true" type="string"
    wssem:modelReference="POOntology#ItemCode"/>
  <element name="itemDesc" nillable="true" type="string"
    wssem:modelReference="POOntology#ItemDesc" />
</all>
</complexType>
```

WSDL complex type element



OWL ontology



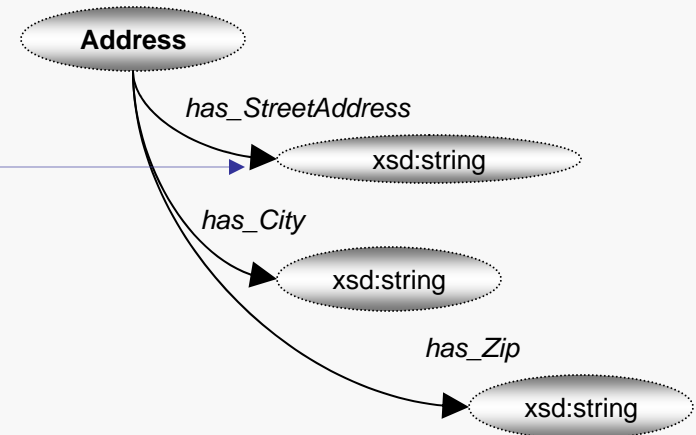
Representing mappings

```

<complexType name="POAddress"
wssem:schemaMapping="http://www.ibm.com/schemaMapping/POAdd
ress.xml#input-doc=doc("POAddress.xml")">
<all>
<element name="streetAddr1" type="string" />
<element name="streetAddr2" 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>

```

WSDL complex type element



OWL ontology

Mapping using XSLT

```

....
<xsl:template match="/">
<POOntology:Address rdf:ID="Address1">
<POOntology:has_StreetAddress rdf:datatype="xs:string">
<xsl:value-of select="concat(POAddress/streetAddr1,POAddress/streetAddr2)"/>
</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>....

```



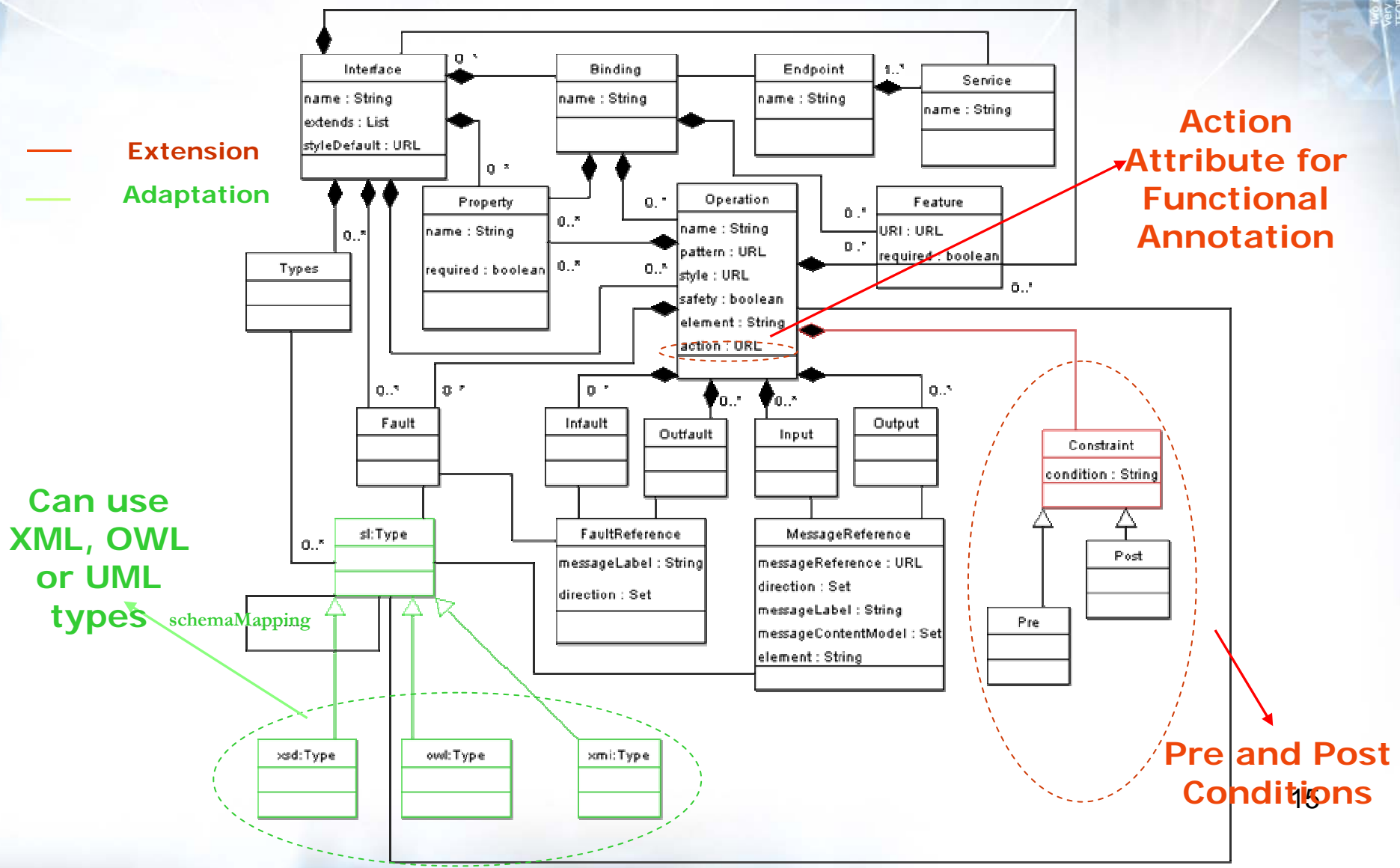
Top of the most active research
Very high build upon research
TEORIS/ Seminars/ Web Services

WSDL-S in perspective



IBM, the most active research... Very much build upon research... TEORIS/2 Seminars, Web Serv...

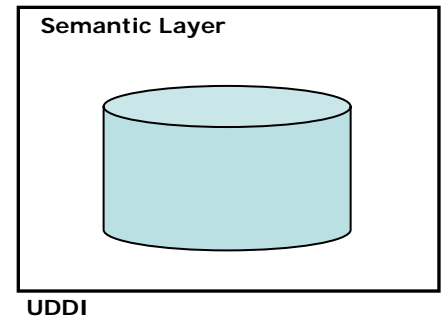
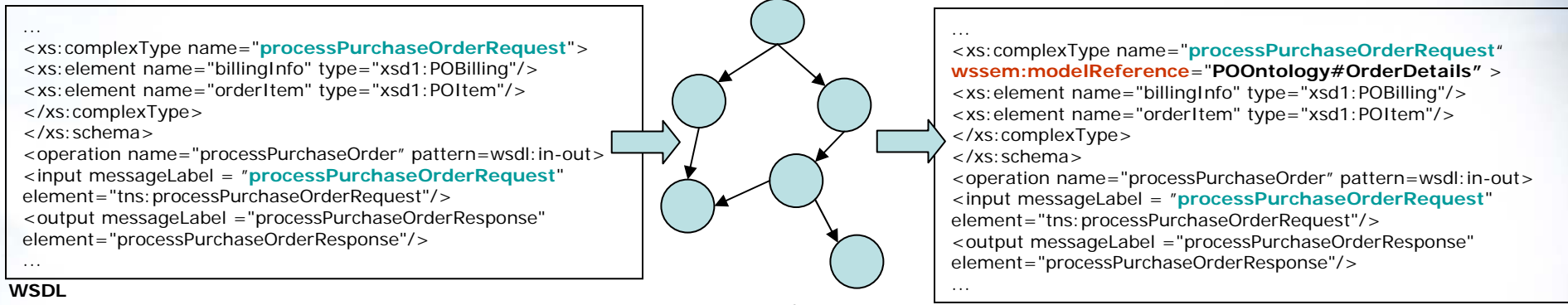
WSDL-S evolution



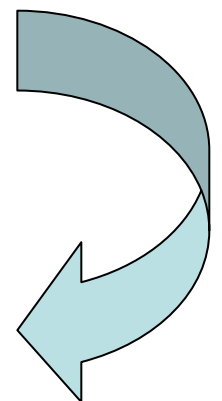


One of the most active research areas in the world today is the development of Web Services. This is a very exciting time for the industry and for the research community. The IEEE Computer Society is sponsoring a series of workshops on Web Services.

WSDL-S in the life cycle of a Web service



UDDI

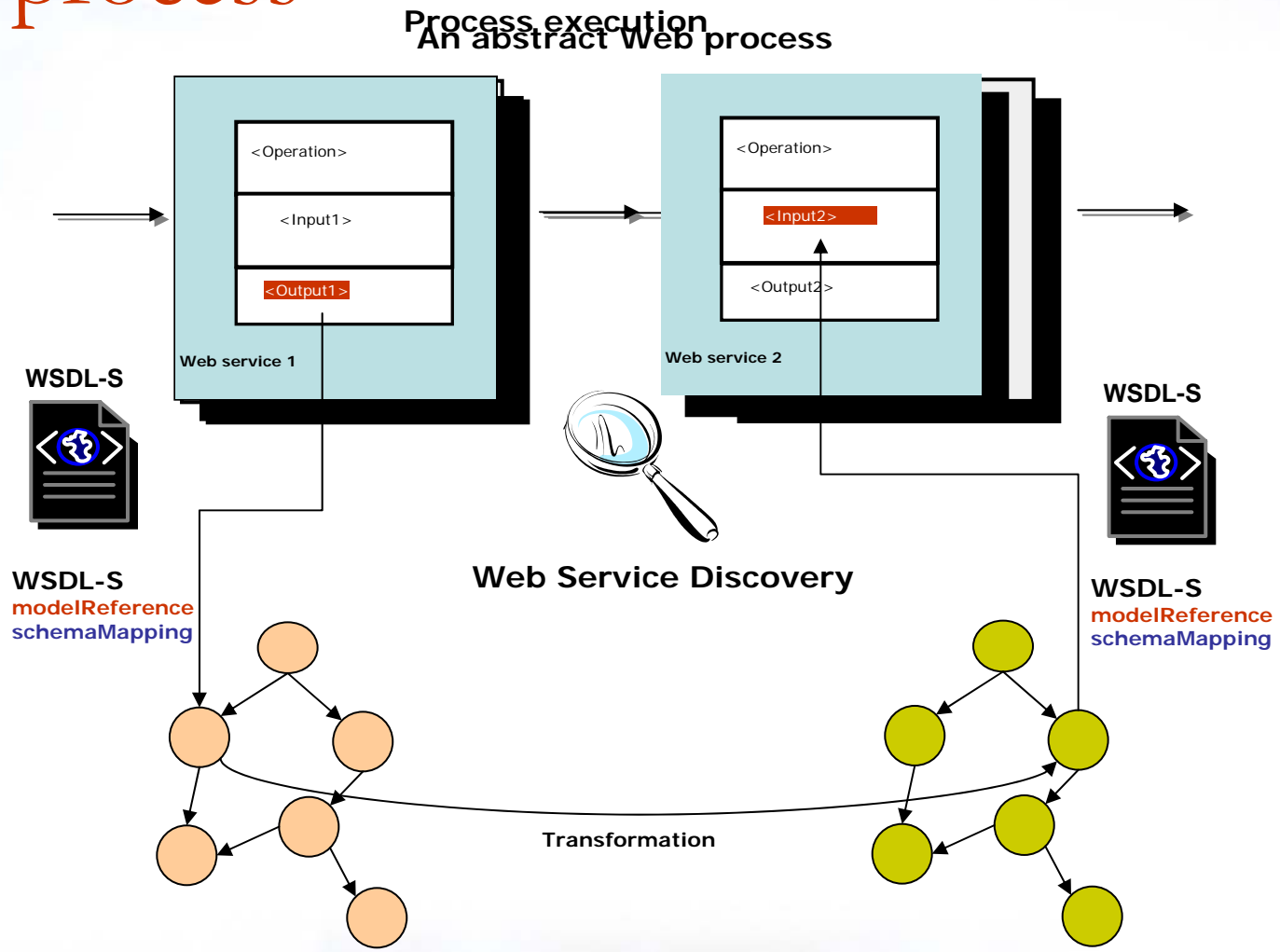


Publishing a service



IBM, the most active research organization in the world, will build upon research in TEORIS, Semantic Web Services

WSDL-S in the life cycle of a Web process



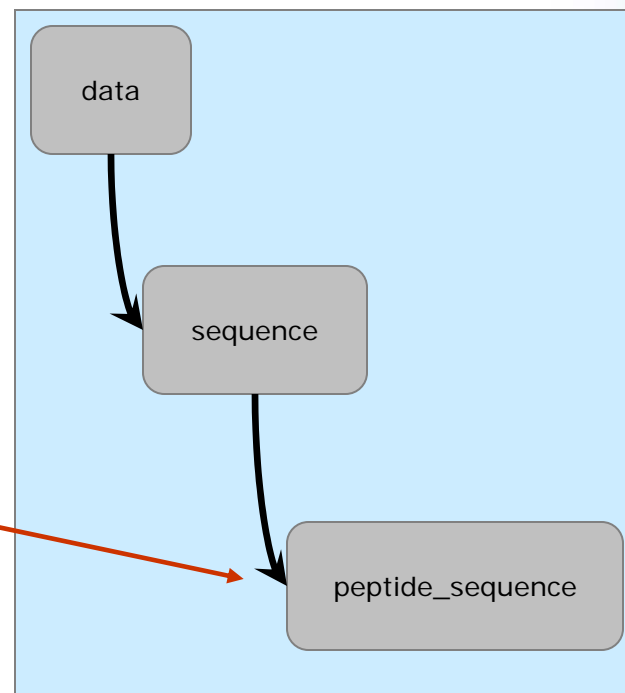


WSDL-S in action

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

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:ngp"
.....
xmlns:wssem="http://www.ibm.com/xmlns/WebServices/WSSemantics"
xmlns:ProPreO="http://lstdis.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>
.....
```

Excerpt: Bio-informatics Web service WSDLS



Excerpt: ProPreO – process ontology



WSDL-S collaborations

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



- Influencing OASIS / W3C



WSDL-S annotator - Radiant

The screenshot shows the Radiant Eclipse IDE with the following components:

- Outline:** A tree view on the left showing the structure of the WSDL file. The `getQuoteRequest` element is circled in red.
- Main Editor:** Displays the WSDL code for `purchaseOrder.wsdl`. The `wssem:modelReference` annotation on line 42 is circled in red: `wssem:modelReference="Ontology0#QuoteRequest"/>`.
- Ontology Navigator:** A panel on the right showing a hierarchical ontology. The `QuoteRequest` class is circled in red, and a context menu is open over it, showing options like `ModelReference`.

```
27 name="statusQuestions" type="xsd:string"/>
28 </message>
29 <message name="getStatusResponse">
30 <part xmlns:LSDISExt="http://lstdis.cs.edu/METEORS/WSDLExtensions"
31 LSDISExt:onto-concept="rosetta:PurchaseOrderStatusResponse" name="r
32 type="xsd:string"/>
33 </message>
34 <portType xmlns:LSDISExt="http://lstdis.cs.edu/METEORS/WSDLExtensions
35 LSDISExt:BusinessEntity="PREE_and_CO" LSDISExt:Category=""
36 LSDISExt:Description="" LSDISExt:GeographicLocation="UGA"
37 name="Annotated_PurchaseOrder">
38 <operation LSDISExt:onto-concept="rosetta:RequestQuote"
39 LSDISExt:operation-expose="true" name="getQuote">
40 <wssem:precondition name="QuoteRequest18171262" wssem:modelReference
41 <input message="tns:getQuoteRequest"
42 wssem:modelReference="Ontology0#QuoteRequest"/>
43 <output message="tns:getQuoteResponse"/>
44 </operation>
45 <operation LSDISExt:onto-concept="rosetta:QueryOrderStatus"
46 LSDISExt:operation-expose="true" name="getStatus">
47 <input message="tns:getStatusRequest"/>
48 <output message="tns:getStatusResponse"/>
49 </operation>
50 </portType>
51 <binding name="PurchaseOrderBinding" type="tns:Annotated_PurchaseOr
52 <soap:binding style="rpc"
53 transport="http://schemas.xmlsoap.org/soap/http"/>
54 <operation name="getQuote">
55 <soap:operation soapAction="" style="rpc"/>
56 <input>
57 <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/
58 namespace="http://127.0.0.1:8080/axis/PurchaseOrder.jws?wsdl"
59 use="encoded"/>
60 </input>
61 <output>
```



Top of the most active research
Very high build upon research
TEOR-S / Semantic Web Service

Why WSDL-S ?

- Approach simple, light-weight, upwardly compatible with the existing WSDL standard
 - practical for adoption
- Approach agnostic to semantic representation language
 - reuse of domain models
 - flexibility in choice of modeling language
 - annotation with multiple ontologies
- Ease in tool upgrades
 - e.g. wsif / axis invocation