LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

Web Service Semantics-WSDL-S

Meenakshi Nagarajan

for the
WSDL-S team

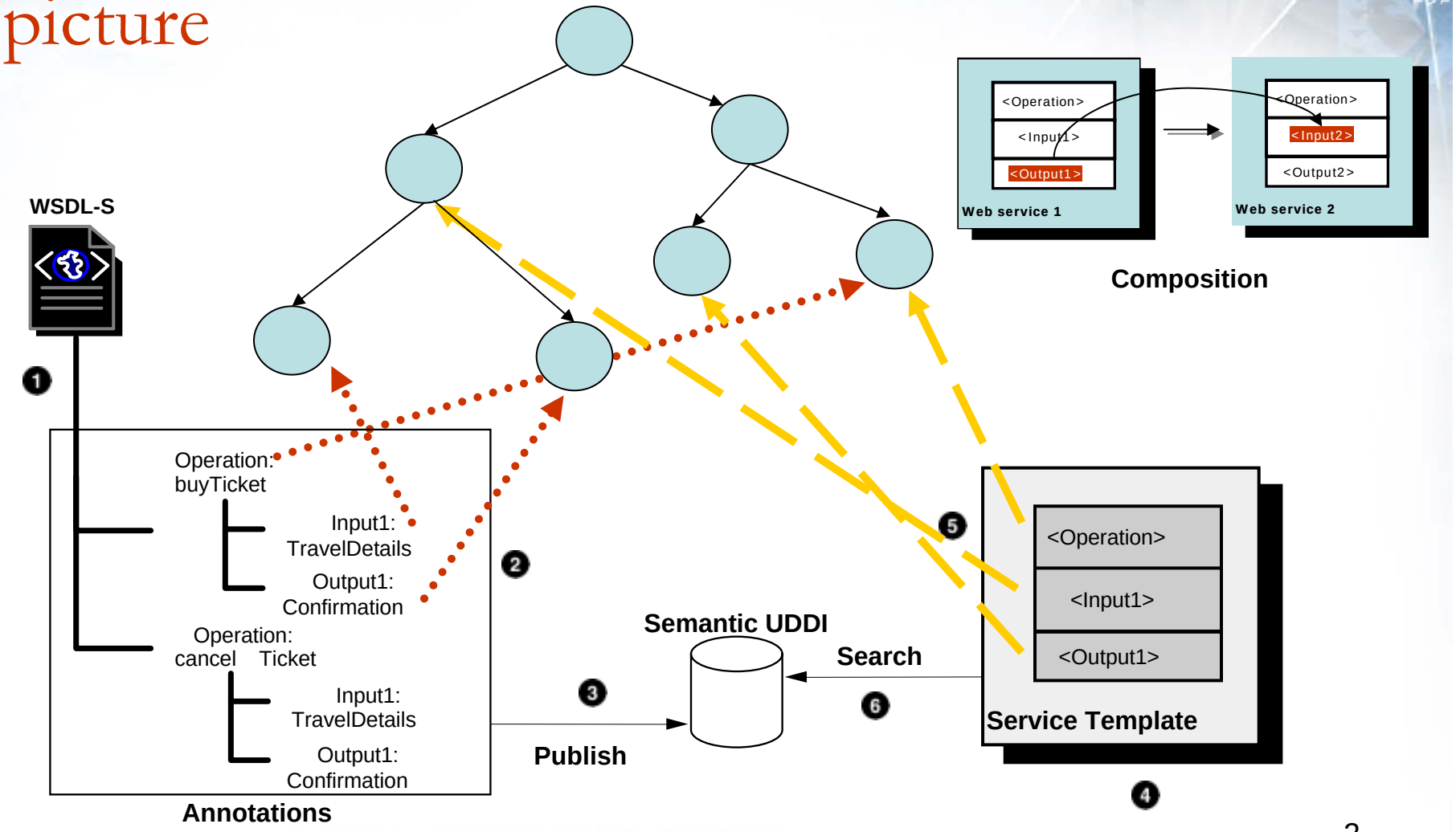
R. Akkiraju*, J. Farrell*, J.Miller, M. Nagarajan, M. Schmidt*, A. Sheth, K. Verma

"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

WSDL-S : scope, proposal and the bigger picture



Sivashanmugam, K., Verma, K., Sheth, A., Miller, J., Adding Semantics to Web Services Standards, ICWS 2003

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
- Provide support for rich mapping mechanisms between Web Service schema types and ontologies

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)

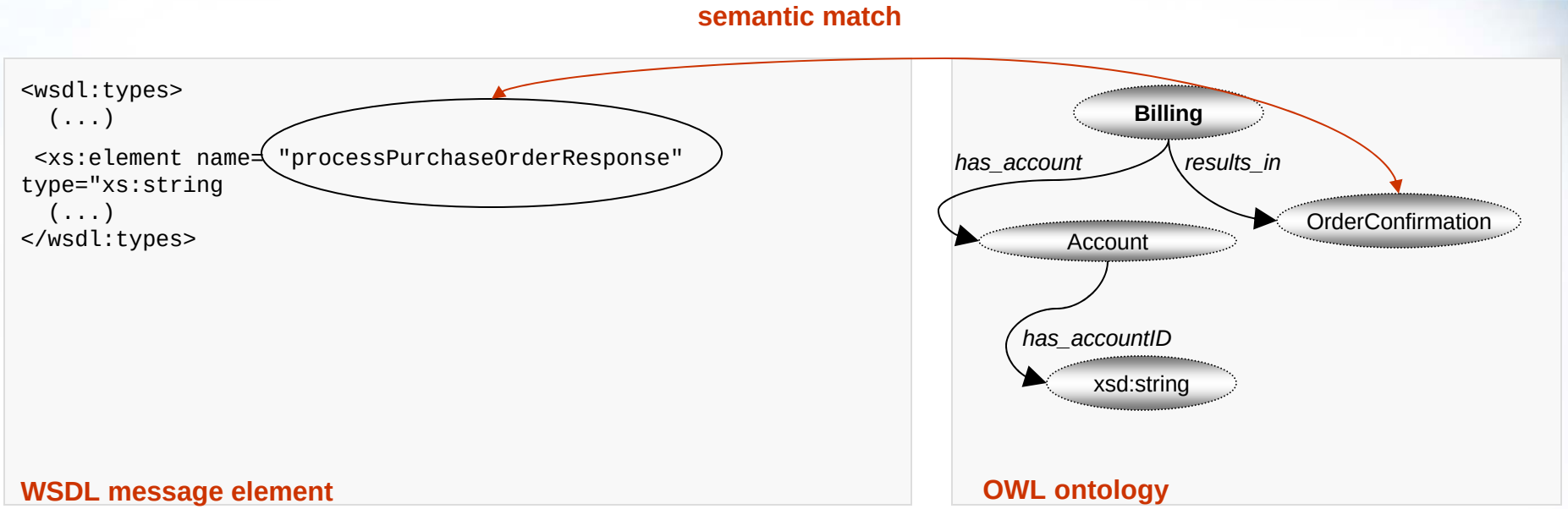
PurchaseOrder.wsdls

```
.....
<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”

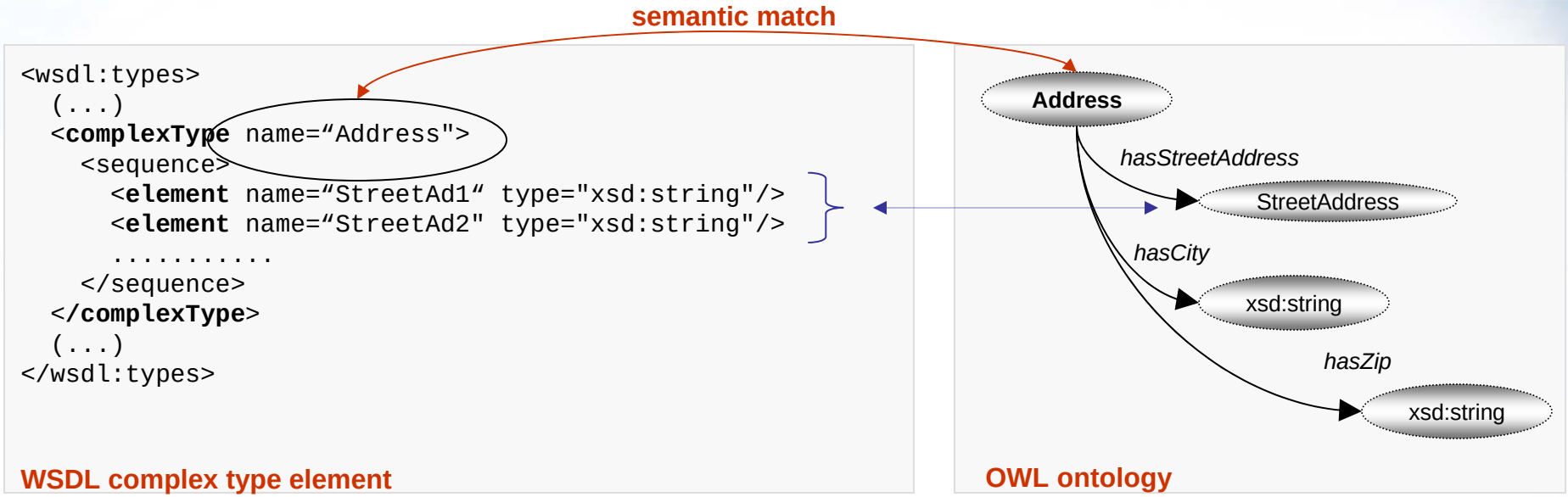
Annotating message types - simple correspondences



1:1 Correspondences

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

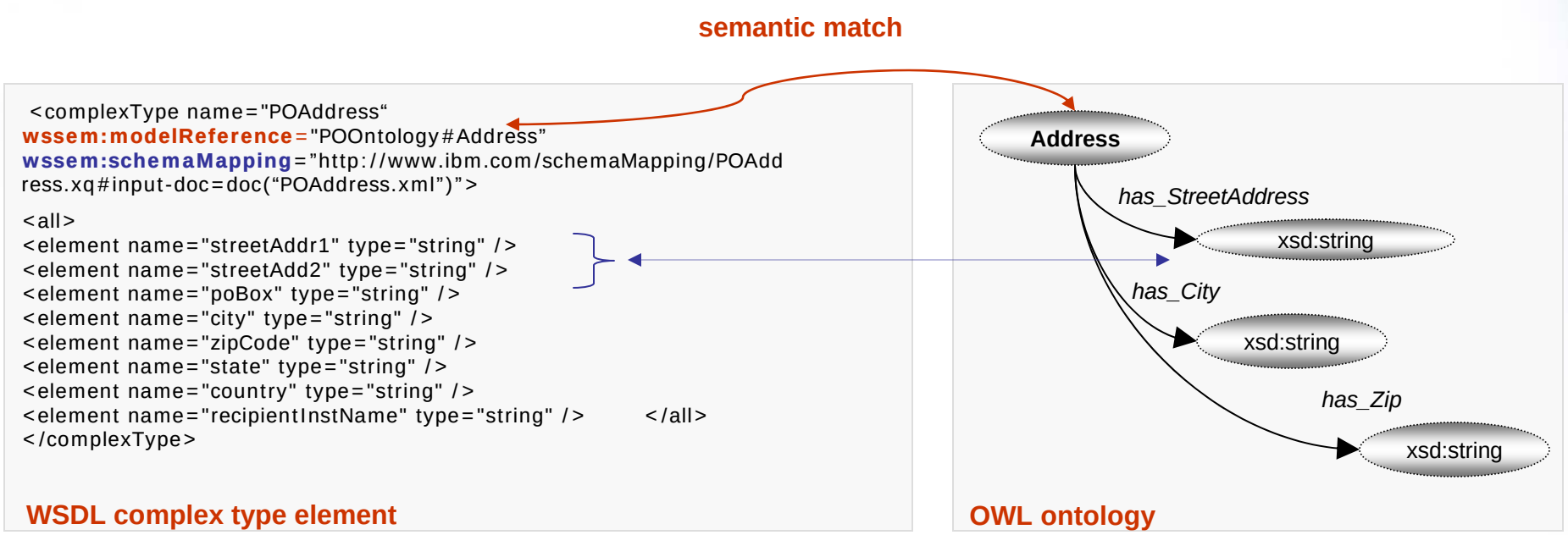
Annotating message types - complex correspondences



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

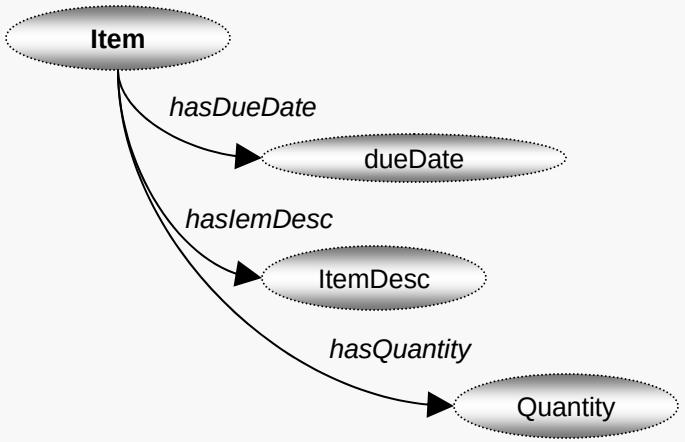


Using modelReference and schemaMapping

- **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

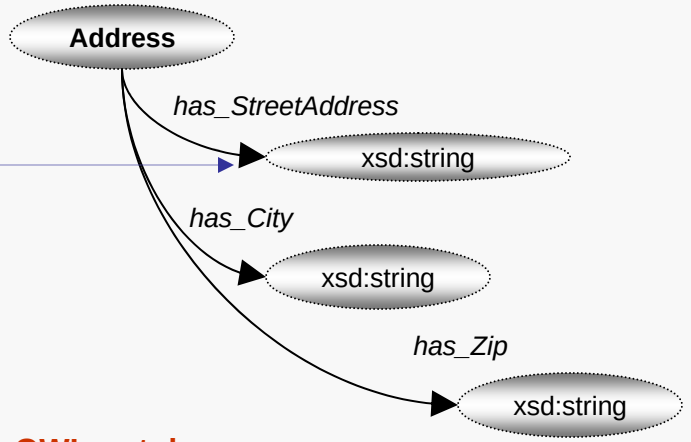


OWL ontology

Representing mappings

```
<complexType name="POAddress"
wssem:schemaMapping="http://www.ibm.com/schemaMapping/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>
```

WSDL complex type element



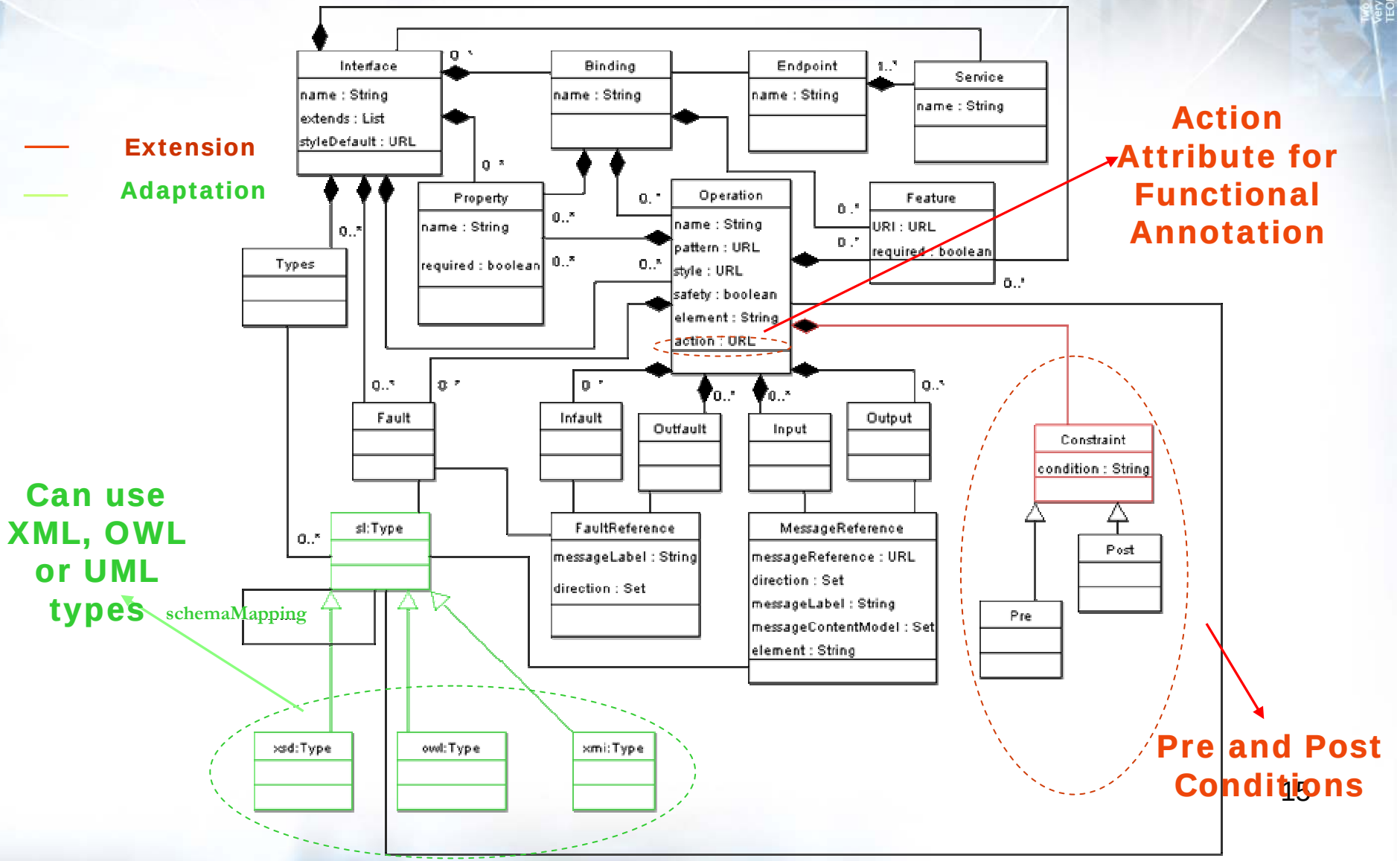
OWL ontology

Mapping using XSLT

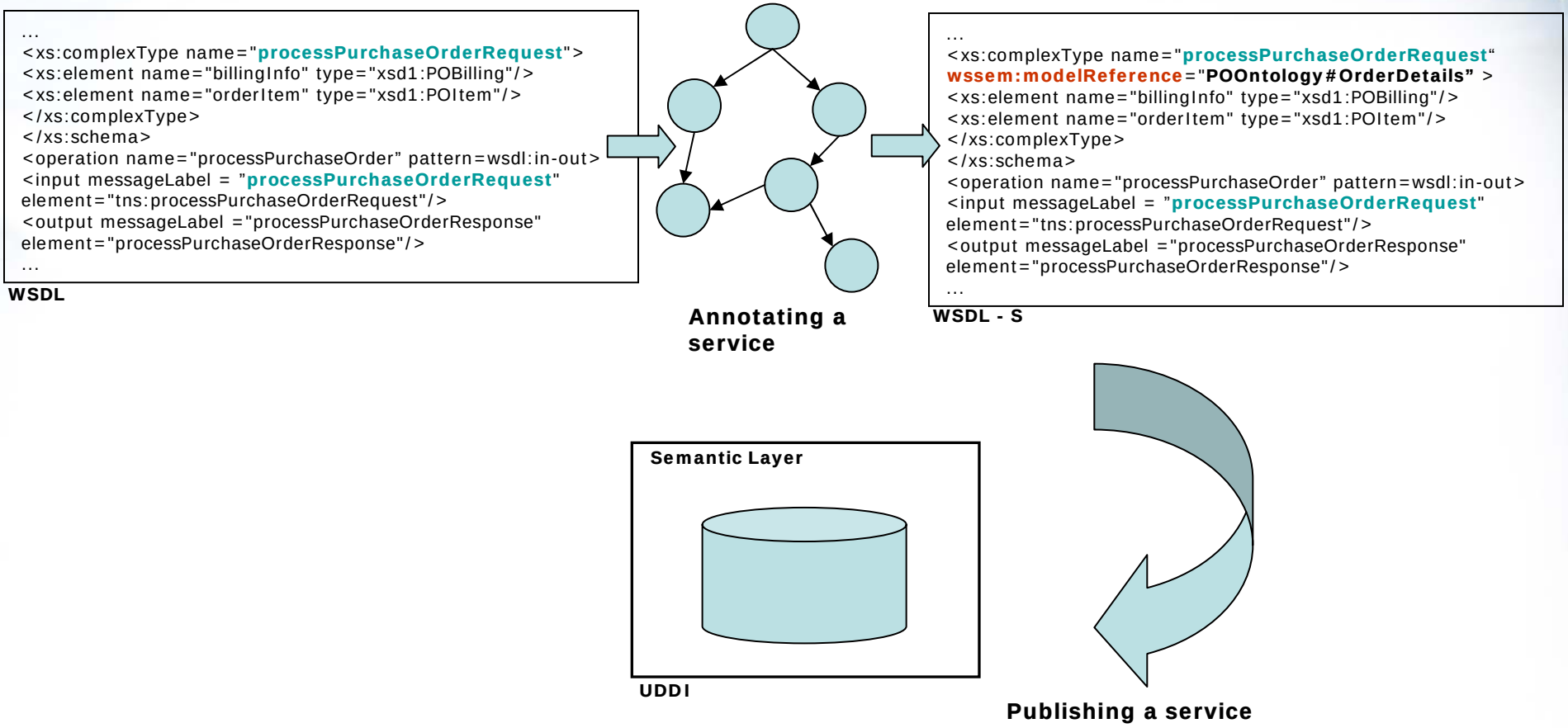
```
....
<xsl:template match="/">
<POOntology:Address rdf:ID="Address1">
<POOntology:has_StreetAddress rdf:datatype="xs:string">
<xsl:value-of select="concat(POAddress/streetAddr1,POAddress/streetAddr2)"/>
</POOntology:has_StreetAddress >
<POOntology:has_City rdf:datatype="xs:string">
<xsl:value-of select="POAddress/city"/>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

WSDL-S in perspective

WSDL-S evolution

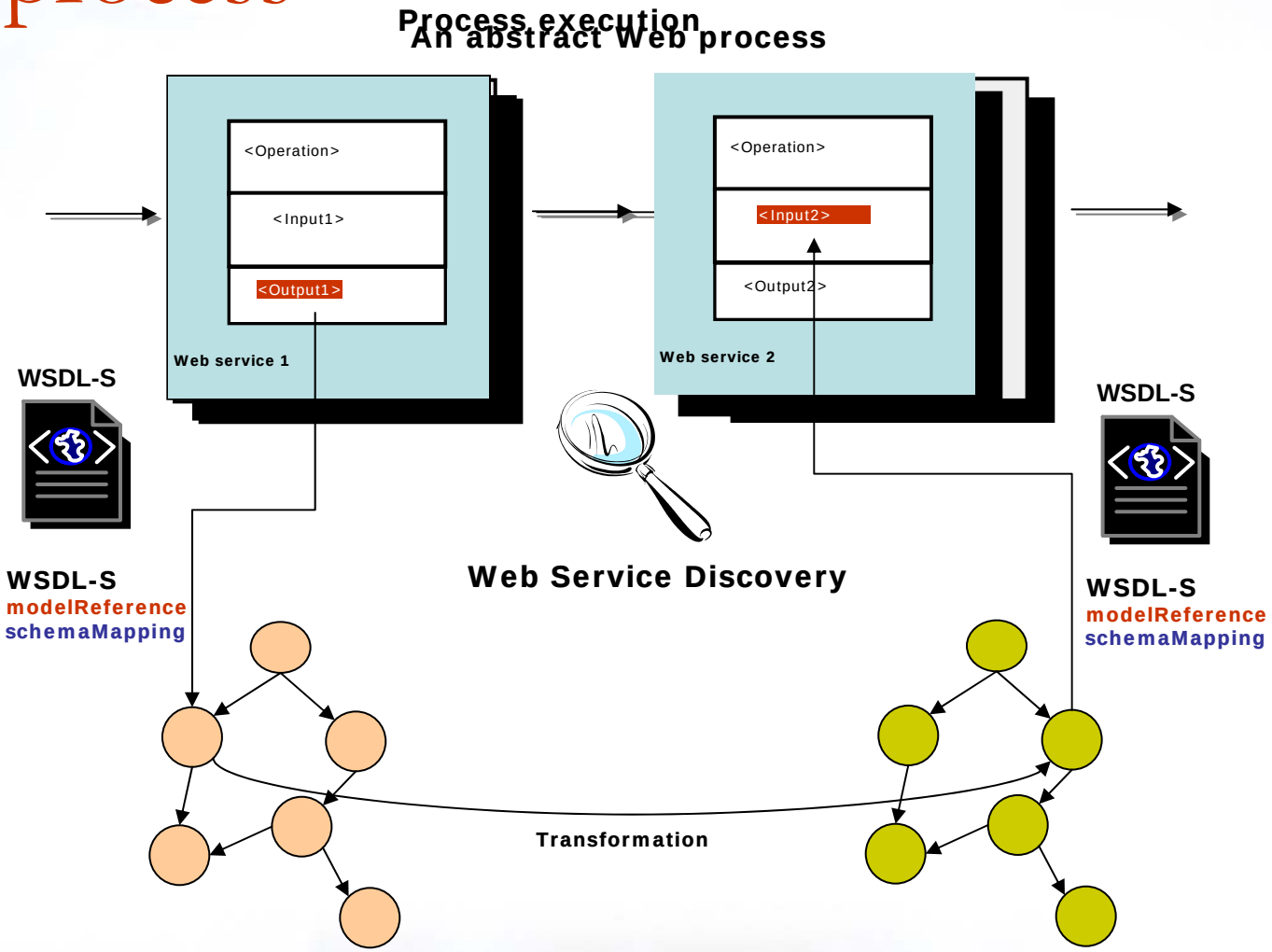


WSDL-S in the life cycle of a Web service



Sivashanmugam, K., Verma, K., Sheth, A., Miller, J., Adding Semantics to Web Services Standards, ICWS 2003

WSDL-S in the life cycle of a Web process



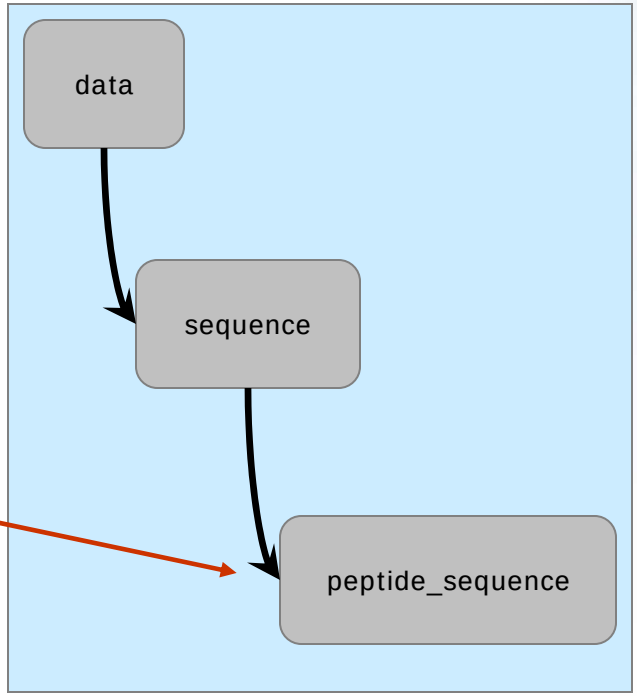
Sivashanmugam, K., Verma, K., Sheth, A., Miller, J., Adding Semantics to Web Services Standards, ICWS 2003

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://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>
……
```

Excerpt: Bio-informatics Web service WSDLS



Excerpt: ProPreO – process ontology

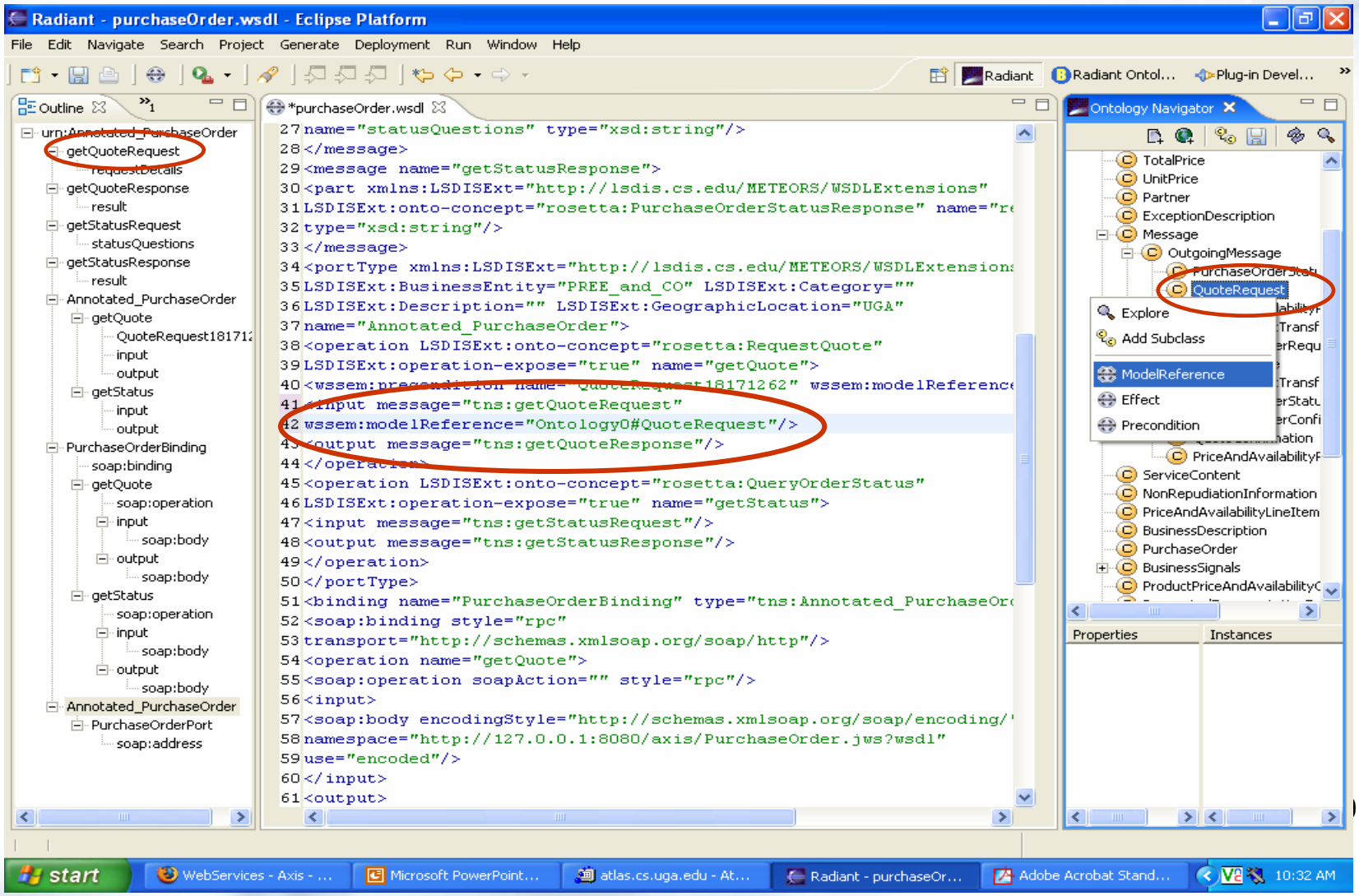
WSDL-S collaborations

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

- Influencing OASIS / W3C

Kunal Verma, Adrian Mochan, Michal Zaremba, Amit Sheth, John Miller, Christoph Bussler, Linking Semantics Web service Efforts - Integrating WSMX and METEOR-S, Second International Workshop on Semantic and Dynamic Web Processes, July 2005

WSDL-S annotator - Radiant



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

More on METEOR-S Web site