LSDIS

Large Scale Distributed Information Systems

University of Georgia
Computer Science Department

# Semantic Interoperability of Web Services – Challenges and Experiences

**Meenakshi Nagarajan**, Kunal Verma, Amit P. Sheth, John Miller, Jon Lathem

meena610@gmail.com
http://lsdis.cs.uga.edu/~meena

LSDIS Lab, Department Of Computer Science, University of Georgia

# Semantic Web services and processes at LSDIS

Meteor-S

Operation Preconditions and Effects

Interoperability – data mediation

Semantic publishing

QoS specifications

Semantic WS – WSDL-S

Policy SLA

Semantic Discovery

Semantic Authentication

Semantics in the entire life cycle

Autonomic Computing

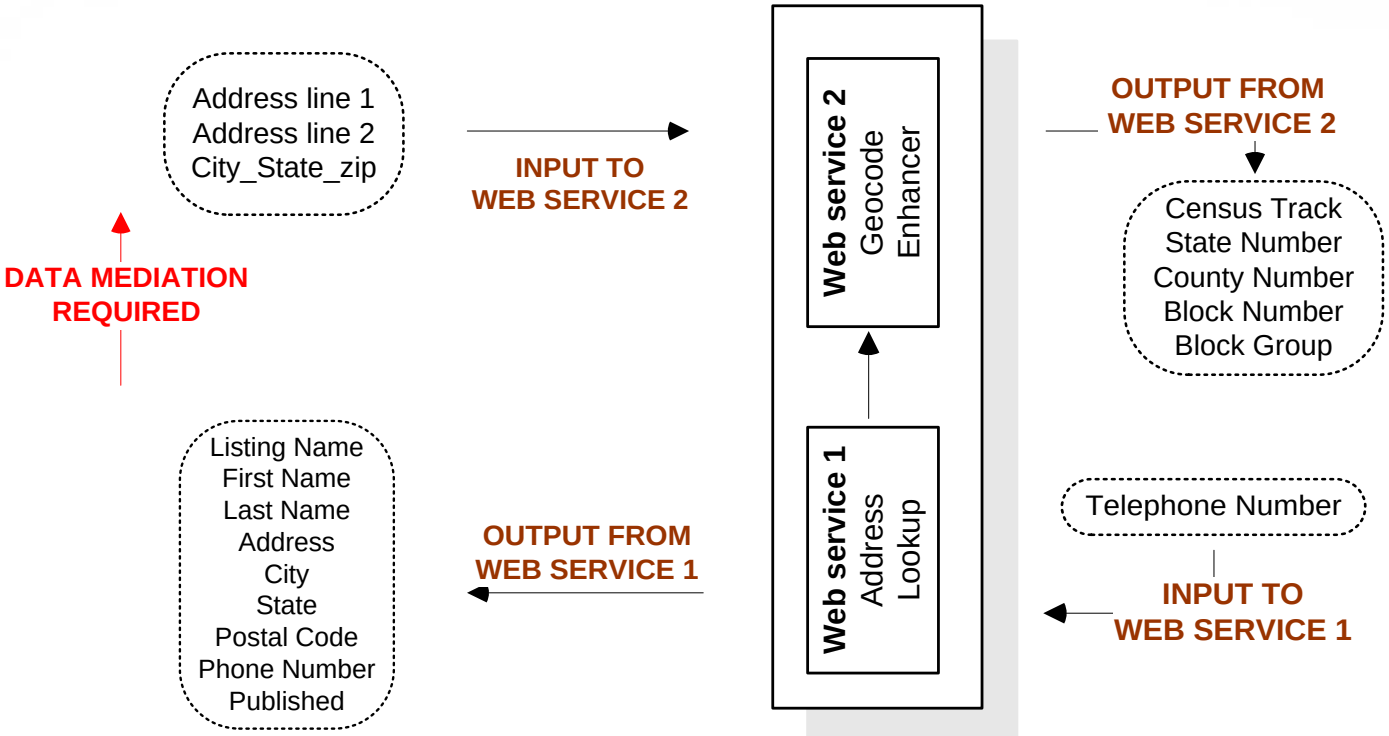
Semantic Composition

Transactions for Web processes

# Web services – then and now

- Surpassed communication, location, system level heterogeneities
- Heterogeneity in structure and semantics continued to exist
- Shifting focus on ‘semantic’ descriptions of service message elements and functionalities
  - Enable automation of discovery, composition, execution etc.
- Semantic match however does not ensure interoperation

# A semantic match alone does not suffice



# Outline of this talk

- Message level heterogeneities hindering interoperation
  - what are they, why do they exist
  - characterizing the different types
- Resolving such heterogeneities
  - State of the art
  - Our approach
- Semantic Web services : WSDL-S
- Using existing WS technology to achieve (automatic) mediation
  - WSDL + Axis 2

# Message level Heterogeneities

# More on Heterogeneities

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

- Web services
  - XML based environment eliminates syntactic and model heterogeneity
  - Structural and Semantic heterogeneities continue to exist

* Semantic and Schematic Similarities between Database Objects: A Context-based approach : Vipul Kashyap, Amit Sheth - VLDB Journal: Very Large Data Bases 1996

# Classifying heterogeneities - 1

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

**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)

***Web service 1***
Student(Id#, Name)

***Web service 2***
Student(SSN, Name)

***Web service 1***
Student(Id#, Name)

***Web service 2***
Book (Id#, Name)

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

***Web service 1***
Student(Id#, Name)
Id# defined as a 4 digit number

***Web service 2***
Student(Id#, Name)
Id# defined as a 9 digit number

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

***Web service 1***
Marks 1-100

***Web service 2***
Grades A-F

# Classifying heterogeneities - 2

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

**Naming conflicts**

**S**emantically alike entities might have different names (synonyms)

Semantically unrelated entities might have the same names (homonyms)

***Web service 1***
EMPLOYEE (Id#, Name)

***Web service 2***
WORKER (Id#, Name)

**Schema Isomorphism conflicts**

Semantically similar entities may have different number of attributes

***Web service 1***
PERSON (Name, Address, HomePhone, WorkPhone)

***Web service 2***
PERSON (Name, Address, Phone)

# Classifying heterogeneities - 3

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

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

***Web service 1***
GRAD-STUDENT (ID, Name, Major)

***Web service 2***
STUDENT(ID, Name, Major, Type)

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

***Web service 1***
PROFESSOR (ID, Name, Dept)

***Web service 2***
FACULTY (ID, ProfID, Dept)

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

***Web service 1***
COURSE (ID, Name, Semester)

***Web service 2***
DEPT( Course, Sem, .., ..)

# Resolving message level heterogeneities

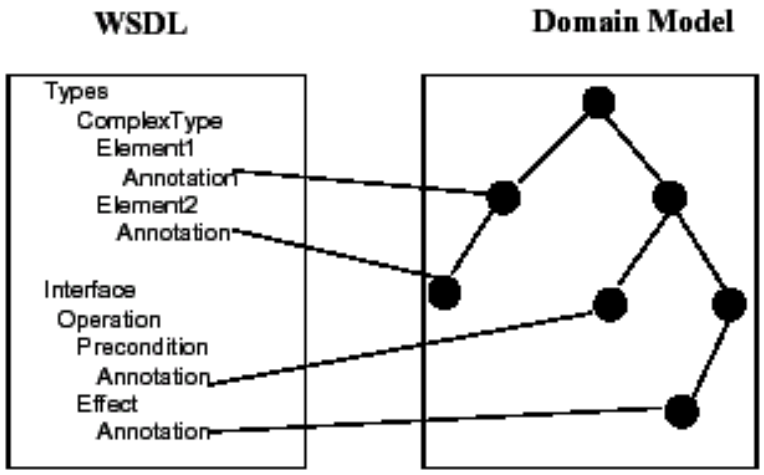
- State of the art solution: Service to service mappings
  - Proposed by most enterprise integration solutions
- Alternate solution: Mapping to a domain semantic model and re-use those to interoperate between services
  - Our approach
- Contributions
  - Comprehensive, practical approach to resolve message / data level heterogeneities
  - Solution borrows from the field of schema/data integration in federated databases
  - A data mediation architecture using extensible elements of WSDL and Axis 2

# Our approach

- Leverages the semantic annotation framework provided by WSDL-S to create data mappings

- WSDL-S
  - Semantics in the entire life cycle of Web services
  - Evolutionary and compatible upgrade of existing Web services standards WSDL
  - Externalize the semantic domain models - agnostic to ontology representation languages.
  - W3C member submission
  - Semantic Annotations for Web Services Description Language Working Group - SAWSDL
    http://www.w3.org/2002/ws/sawsdl/

# WSDL-S




- 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 element : action (under consideration) (on operation element)

# PurchaseOrder.wsdls

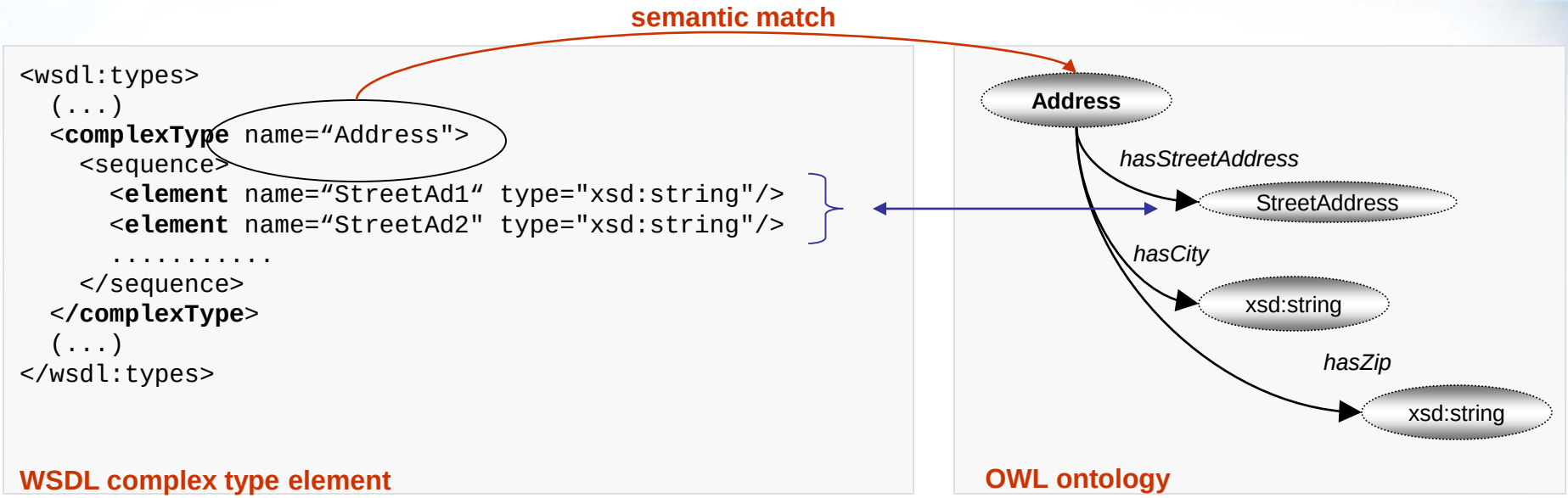
.....

```
<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>
<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 Message elements




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

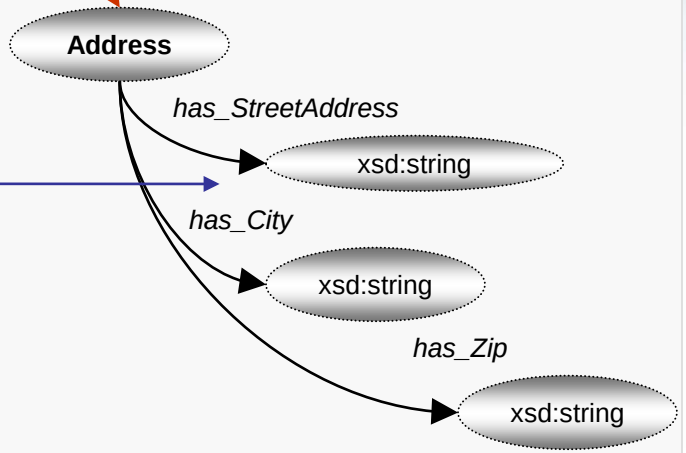
# Example Annotation

**semantic match**

```
<complexType name="POAddress"
wssem:modelReference="POOntology#Address"
wssem:schemaMapping="http://www.ibm.com/schemaMapping/POA
ddress.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**




**OWL ontology**

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

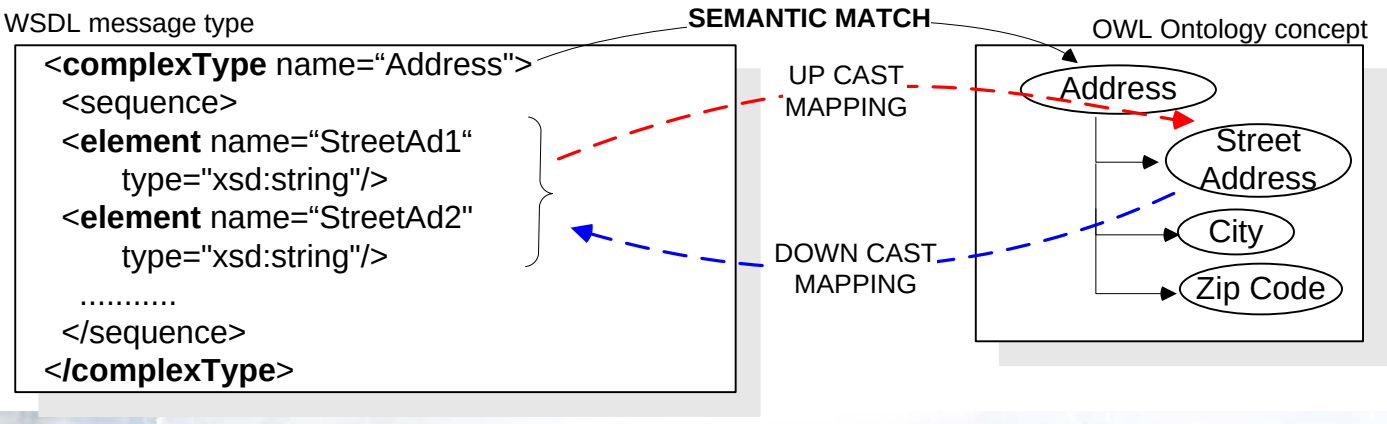
# Want to know more about WSDL-S ?

- W3C submission Web page
  http://www.w3.org/Submission/WSDL-S/
- Project and related tools (annotation tools)
  http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/
- Presentation at W3C Workshop on Frameworks for Semantics in Web Services
  http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/WSDL-S.pdf
- OR Talk to me!

# Resolving message level heterogeneities using WSDL-S

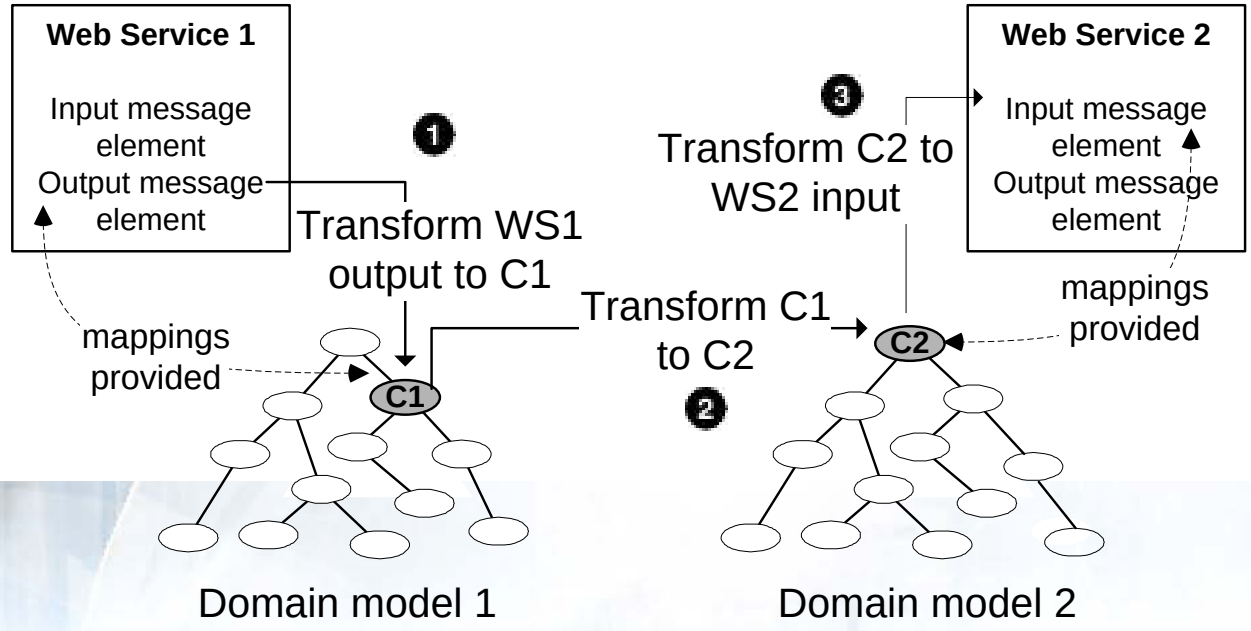
# WSDL-S support for data mediation

- 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



# Realizing data mediation

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



# Data Mediation System Architecture

- Focus: Easy incorporation of tooling support for SWS in existing tools
- Uses extensibility support offered by WSDL and Axis 2 (handlers)
- Pre-requisites
  - Web services should be described using WSDL-S
  - The upcast and downcast mappings from the Web service message elements to the semantic concepts should be created
  - The Web services must be deployed and the WSDL-S files must be accessible. Axis 2 allows deployment of WSDL-S files.

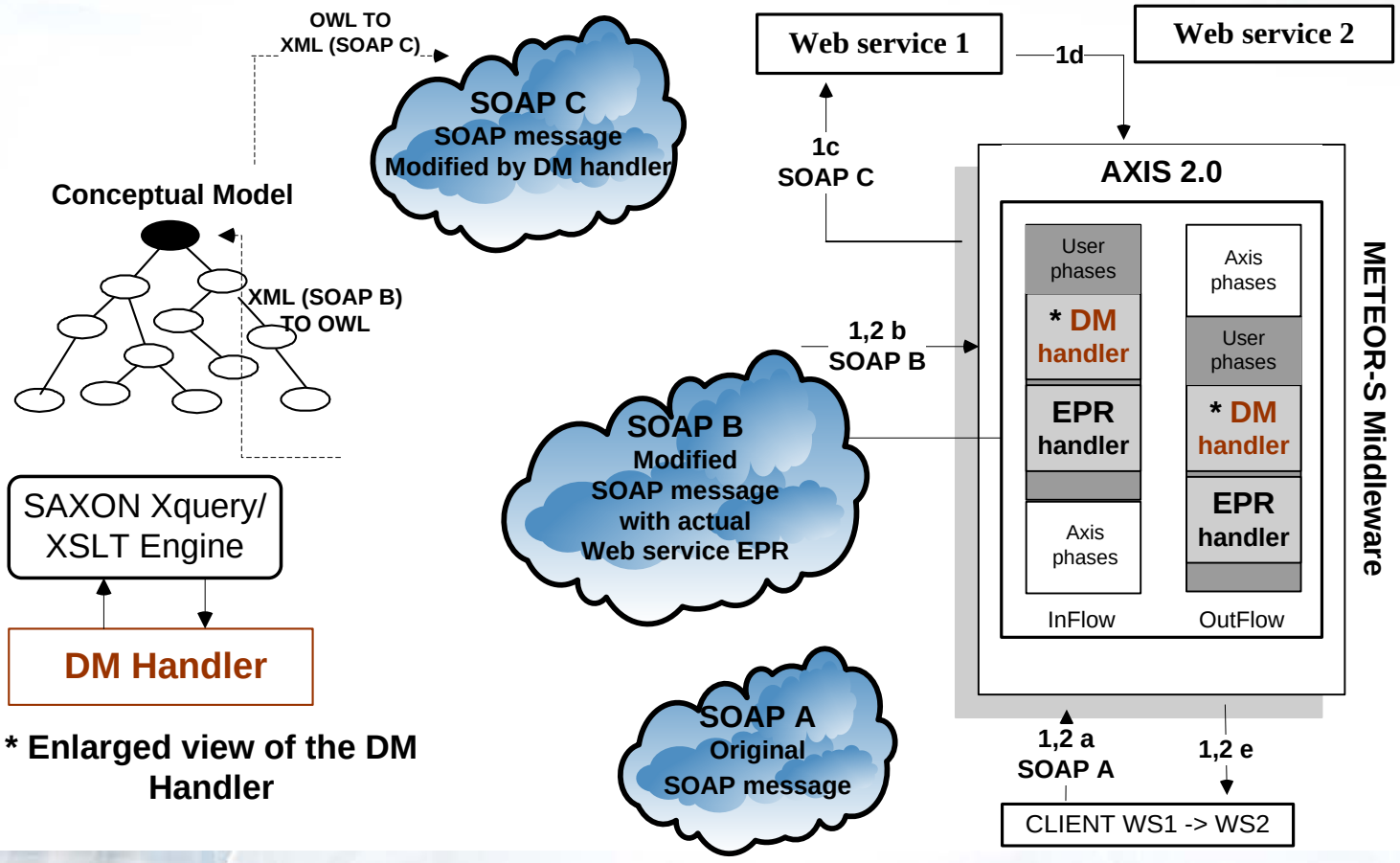
# DM Architecture components

- Part of the 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)

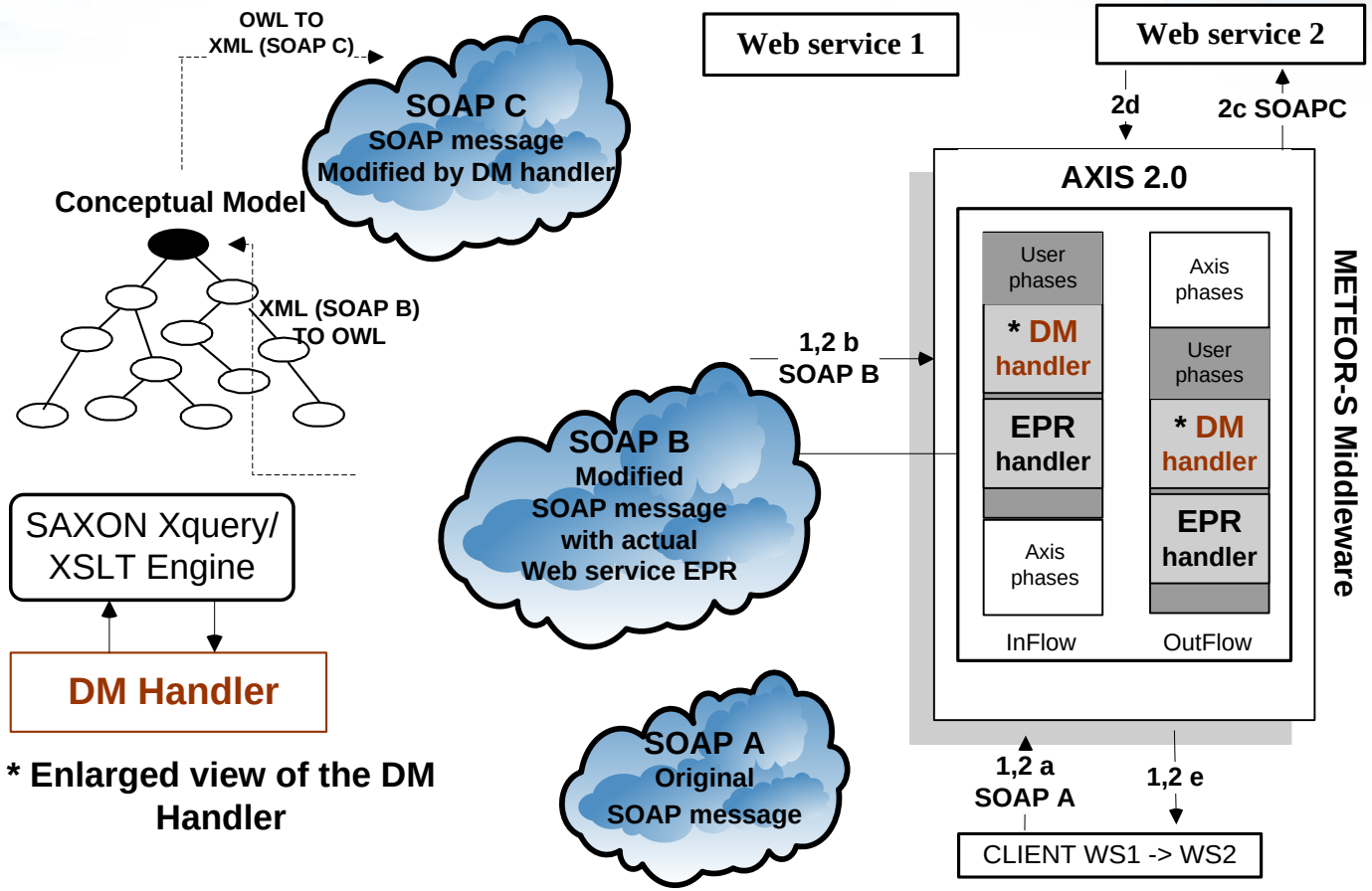
# DM Handler – a closer look

- Each time a Web service is invoked
  - obtains the *'schemaMapping*' functions from WSDL-S locations (using the WSDL-S4J API)
  - performs the up cast and downcast mappings on the incoming SOAP message using a mapping processor/engine (SAXON for XQuery and XSLT)
  - updates the SOAP message. Appropriate Axis handlers then invoke the Web service with the transformed message.

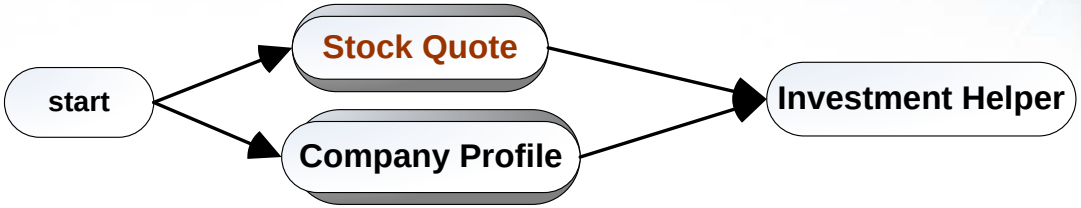
# Walk through example – WS1 invocation



# Walk through example – WS2 invocation



# Evaluation




| URI of stock quote Web services | Message-level Heterogeneities | | | | | Can achieve interoperability using mappings |
|---|---|---|---|---|---|---|
| | Structural | Schema Isomorphism | Attribute Naming | Entity Naming | Data Repres. | |
| http://ws.strikeiron.com/SwanandMokashi/StockQuotes?wsdl | Yes | Yes | Yes | Yes | Yes | Yes |
| * http://ws.strikeiron.com/HistoricalStockQuotes?wsdl | Yes (minor) | Yes | No | Yes | No | Yes |
| http://ws.strikeiron.com/BasicRealTimeQuotes?wsdl | Yes | Yes | Yes | Yes | Yes | Yes |
| * http://www.webservicex.net/stockquote.asmx?wsdl | Yes (minor) | Yes | Yes | Yes | No | Yes |
| http://www.xmethods.net/sd/StockQuoteService.wsdl | Yes | Yes | Yes | Yes | Yes | Yes |
| glkev.webs.innerhost.com/glkev_ws/StockServices.asmx?wsdl | Yes | Yes | Yes | Yes | Yes | Yes |
| www.gama-system.com/webservices/stockquotes.asmx?wsdl | Yes | Yes | Yes | Yes | No | No |
| glkev.webs.innerhost.com/glkev_ws/HistoricalStockQuotes.asmx?wsdl | Yes | Yes | Yes | Yes | Yes | Yes |
| ws.cdyne.com/delayedstockquote/delayedstockquote.asmx?wsdl | Yes | Yes | Yes | Yes | Yes | Yes |
| * www.xignite.com/xquotes.asmx?WSDL | Yes (minor) | Yes | Yes | Yes | No | Yes |

**All stock quote Web services to interoperate with investment helper service – available at http://lsdis.cs.uga.edu/~meena/ICWS06/Eval.html**

*** These services could interoperate with the investment helper service using very minor mappings between the message schemas**

# Discussion

- Matching and mapping are hard problems
  - That is not what we claim to solve
- Need for a light weight semantic framework for Web services – WSDL-S
  - Simply extending this to achieve complete actual interoperation
- Interoperation in a multiple ontology environment
  - Inter ontology matching and mapping

# Conclusion

- Comprehensive solution for resolving message level heterogeneities
  - Extending available semantics to pre-define message level mappings
  - Extending the state of the art
- Data mediation is a hard problem
  - WSDL-S and this work is an important first step