Data Mediation to Message Level Conflict in Heterogeneous Web Services

Md Alamgir and Mohd Ghazali Mohayidin
Information Technology, Centre for Graduate Studies, Open University Malaysia
{malam68, mghazalee}@oum.edu.my

***Abstract*- Enterprise Information Systems (EISs) are built in isolated and independent environments leading to unpredictable and incompatible structure of data stores. An EI system can expose its functionalities as Web services to share resources of existing global internet infrastructure. The goal of this research paper is to facilitate inter-operation between Web services. Successful and reliable information (message) exchange between Web services is necessary to meet the current challenge of Enterprise Information Integration (EII). A real-world business process, which consists of Web services WS1 and WS2, can be used as a practical scenario for data or message exchange between Web services. In this scenario, message is exchanged by using output of WS1 as input of WS2. If data format of WS1 and WS2 are heterogeneous or incompatible, interoperation between them is impossible if data mediation is not used to resolve message level conflict and incompatibility in the context of syntax and semantics. Data Mediation requires mapping a message from one format to another. We propose to derive a mediation technique that will enable previously less inter-operative heterogeneous Web services to become more inter-operative now. To improve inter-operational performance between Web services our data mediation approach extends and utilizes existing Web service supporting tools WSDL and SAWSDL.**

***Keywords*- Enterprise Information System, Data Store, Business Process, Data Mediation, Web Service, Mapping, WSDL, SAWSDL**

1. MOTIVATION

Web services have emerged to face the challenging task of interoperating among them to provide integration solutions to Enterprise Information Systems. To utilize and reuse the huge investments and efforts made to build world wide internet infrastructure and software resources of EISs, enterprise software exposes functionalities as Web services. As Web services are implemented in isolated and independent environments, data structure of web services is heterogeneous leading to difficult interoperation. Incompatible data and information are required to transform into one another to insure successful communication and operation between web services. The motive of this paper is to make a research contribution aiming to derive an approach or technique to minimize data or message level heterogeneity between web services. We use two different web services WS1 and WS2. WS1 acts as an address locator which feeds to WS2 to provide weather information. Output from WS1 is used as input to WS2. The structure of the two data sets (output and input) is heterogeneous raising problem for interoperation.

2. INTRODUCTION

The emergence of Web service and its architecture is to provide enterprise solutions by composing Web services to constitute processes in the domains of business and science. One of the main problems faced by interoperation between web services is data heterogeneity. Interoperation between web services can be improved by resolving data heterogeneity. Data mediation is a popular technique applied to solve heterogeneity between web services. Data mediation requires mapping between heterogeneous data sets to a conceptual model and using the mapping for interoperation of web services. Before proceeding to the next sections, some terms and concepts regarding web service, data mediation, matching, and mapping are required to be discussed to help understand the statements in this paper.

2.1 Definitions of Terms

A *Web service* is a software application, accessible on the Web (or an enterprise's intranet) through a URL, that is accessed by clients using XML-based protocols, such as Simple Object Access Protocol (SOAP) sent over accepted Internet protocols, such as HTTP [1]. A Web service exposes its interfaces and bindings for the access of clients to it. Interfaces and bindings of a Web service are defined using XML artifacts, such as *Web Service Definition Language* (WSDL). WSDL is used to describe Web services. *SOAP* describes communication protocol for Web services.

Two web services can have similar descriptions while having different meanings or they can have different descriptions while having the same meaning. And thus descriptions of two Web services can be ambiguous. Resolving such ambiguities in Web services descriptions is an important step toward automating the discovery and composition of Web services [3]. Semantic Annotations for WSDL (*SAWSDL*) defines mechanisms to add semantics to WSDL components to resolve ambiguities in web service descriptions.

Data mediation includes tasks such as data transformation, translation, reconciliation, creation, synthesis, and integration from multiple data sources [2]. *Matching* can be defined as finding similarities between messages elements. *Mapping* can be defined as the physical connectivity between the matched elements and the rules for transforming one schema to another. A *schema* can be defined as a model to describe the structure of information [4]. *Lifting* is the process of semantically annotating a source schema with ontology [6]. *Ontology* is a formal representation of a set of concepts within a domain and the relationships between those concepts. The ontologies now become a vehicle through which Web services resolve their message level heterogeneities [7].

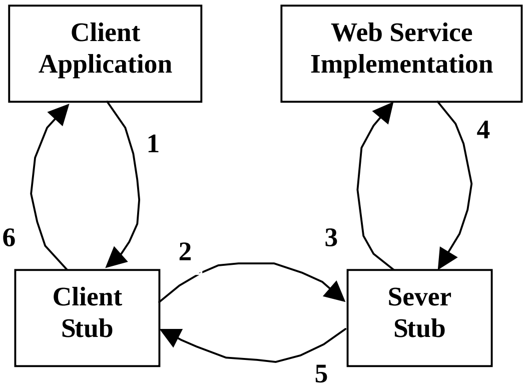

Fig1: Structure of the web service application

As shown in fig1, client and server stubs are generated from the WSDL descriptions. Client stub generates SOAP requests and interprets SOAP responses from the server. Server stub interprets SOAP requests sent by the client and generates SOAP responses. SOAP requests and responses are sent over a network using HTTP protocol. 1: Client stub converts local invocation into SOAP request. 2: SOAP request is sent over the network. Server stub receives the SOAP request and deserializes it. 3: Server stub invokes on the service implementation. 4: Server stub receives the requested results and converts them to the SOAP response. 5: SOAP response is sent back over the network. 6: Client stub deserializes the results from the SOAP message and return them to the client.

2.2 How a Client Invokes a Web Service

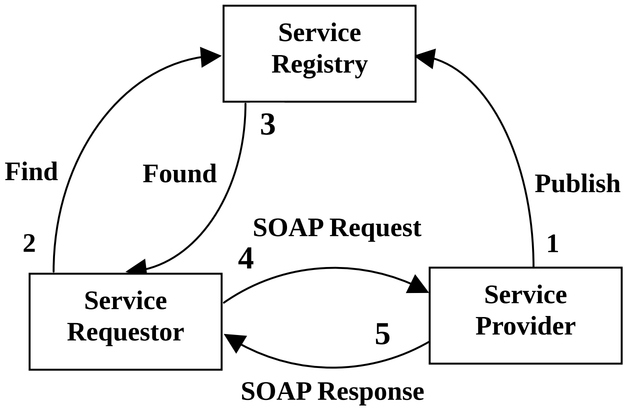

Fig2: How a client invokes a web service

The steps shown in fig2 are the followings. 1: Service provider publishes the web service in a service registry. 2-3: Web service is discovered from the service registry to meet the requirements and the service provider is located. 4: Client gets the WSDL (from the service provider), which describes the web service to generate SOAP request. 5: Web server generates SOAP response to be sent to the client.

2.3 Layered Architecture of Web Service Communication

As shown in fig3, web service is specified by Web Service Description Language (WSDL). When a client needs a service, it reads the WSDL and generates a SOAP message. The SOAP message is transported over the network to the service provider. Following the message instructions, the service is implemented and executed on the server side.

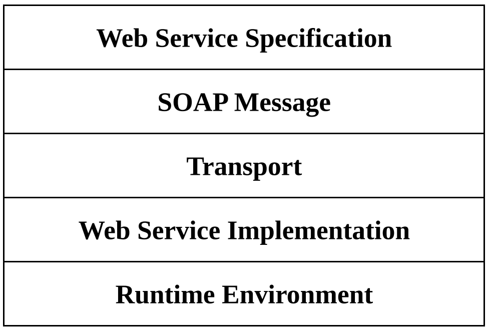

Fig3: Layered Architecture of Web Service Communication

2.4 Server Side of a Web Service

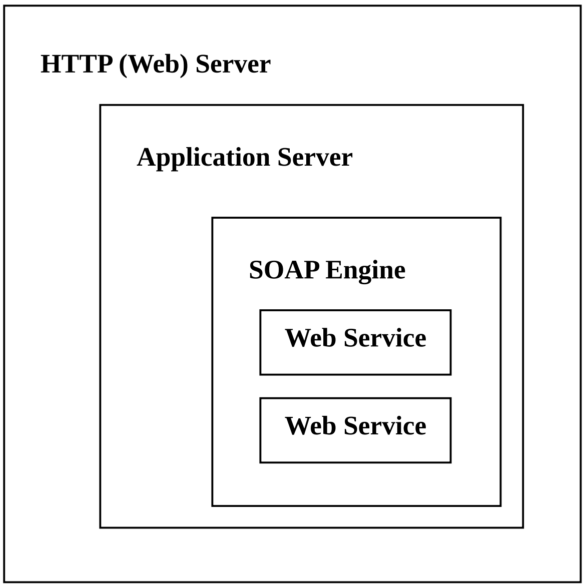

Fig4: Server side in a web service application [5]

SOAP engine is a piece of software that knows only how to handle SOAP requests and responses. Apache Axis an example of SOAP engine. Application server is a piece of software that contains applications to be accessed by different clients.

3. MESSAGE LEVEL CONFLICT

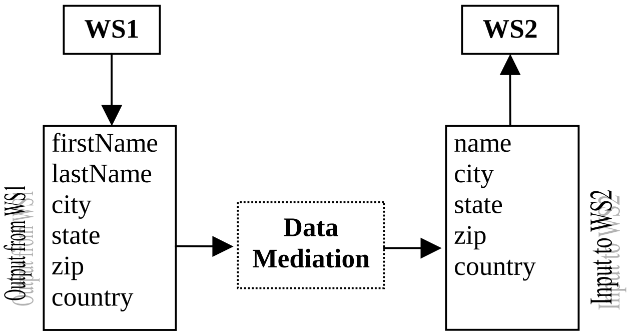

Fig5: Data Mediation between Web Services

As shown in figure 5, output data from WS1 and input data to WS2 are heterogeneous to create a problem for interoperation. On one side (source), first name and last name are separated. On the other side (target), name is combined by concatenating the first and last names. Output from WS1 can be termed as based on source schema. Input to WS2 can be termed as based on target schema. The solution to the problem is to convert the source schema to the target schema through what is called data mediation.

4. PROPOSED SOLUTION

The main task of data mediation is to transform one message to another. Practical situation is that an entity in a message can be described using different terms and different entities can be described using the same term. Therefore, word to word transformation will not produce expected results. The transformation should be based on the context to secure the semantics. The simple solution to get a message transformation is to manually map from source schema to target schema. This mapping is not feasible, because any change in the service or process will require re-mapping. An alternate solution is to map each schema to a conceptual domain model. This conceptual model is known as ontology that acts as a common vocabulary to contain the terms used by the Web services. In this approach, each element of a message is attached to a semantic annotation and those annotations are stored in the ontology. When transforming from one message format to another, the ontology is consulted to get an appropriate matching. Using this matching, a mapping is established between the source and target schemas. These mappings are used for interoperation between Web services.

The proposed solution intends to utilize existing tools: WSDL, SAWSDL, OWL (Web Ontology Language), and Axis 2. WSDL (Web Service Description Language) is used to specify the functionalities of a Web service. SAWSDL (Semantic Annotations for WSDL) is an effort to define mechanisms by which semantic annotations can be added to WSDL components [7]. OWL is used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. Axis 2 provides extensibility support to SAWSDL to implement the data mediation.

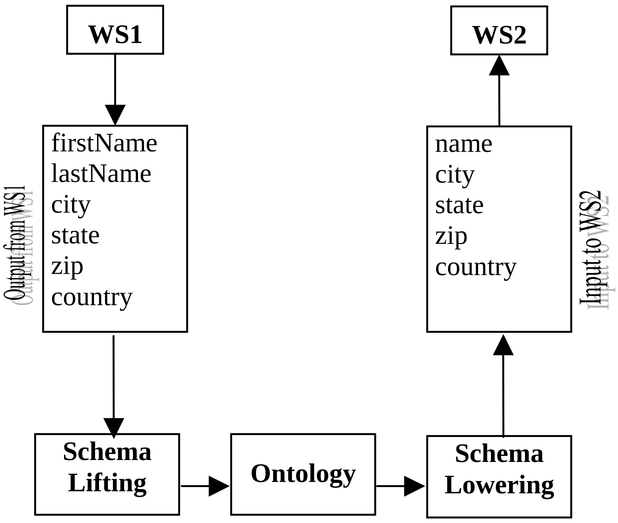

Fig6: Data Mediation Using SAWSDL

As shown in figure 6, Web service WS1 outputs a set data containing six elements (firstName, lastName, city, state, zip, and country) formatted in a XML document as shown in figure 8(a). Web Service WS2 takes a set of input data containing five elements (name, city, state, zip, and country) formatted in a XML document. It is seen that schemas of both set of data are incompatible. Output from WS1 cannot be fed to the input of WS2 directly. Let S_i and S_o be the schemas defined for output from WS1 and input to WS2. The semantic information for elements of both schemas is stored in a common vocabulary or ontology known as a conceptual domain model (M_o). Transformation from S_i to S_o is performed in two steps. 1) Schema S_i is converted to the model M_o through a process known as schema lifting. 2) The model M_o is converted to the schema S_o through a process known as schema lowering.

4.1 Schema Lifting

XML Keywords	**OWL Keywords**
complexType	Class
element having other elements or at least one attribute	Class and ObjectProperty
element without sub-elements or attributes.	DatatypeProperty
simpleType	DatatypeProperty
minOccurs	minCardinality
maxOccurs	maxCardinality
sequence, all	intersectionOf
choice	Combined with intersectionOf, unionOf, and complementOf

Fig7: Correspondence between XML and OWL Elements

S_i schema is mapped to the model M_o by a tool – OWL (Web Ontology Language). OWL is intended for modeling the semantic relationships of a domain [8]. The mapping consists of transforming XML components (or keywords) to OWL components (keywords). Figure 7 shows various elements of XML mapped to the corresponding elements of OWL. An identifier is mapped from XML schema to an URI by concatenating the targetNamespace URI, the character #, and component's local name.

4.2 Converting WSDL to SAWSDL

SAWSDL (Semantic Annotations WSDL) provides a standard means by which WSDL documents can be related to semantic descriptions [10]. Radiant is an eclipse plug-in that provides a user interface for annotating existing WSDL documents using OWL ontology to create a SAWSDL file [7]. SAWSDL defines an extension attribute *modelReference* as shown in figure 8(a) to specify the association between WSDL or XML schema and ontology (semantic model). The attribute modelReference is associated with an ontology identifies by *Address*. As shown in the figure 8(c), the address ontology, which is also an XML document, is defined or created using OWL-an ontology language. The attribute *liftingschemaMapping* defined by SAWSDL is associated with the location of the document representing the semantic model. The location of semantic model is important for information related to web service discovery. Figure 8(b) describes the ontology in graphical format. In the tree-shaped graph, it is seen that the relation between name and {firstName, lastName} can be extracted to have a semantic match to resolve structural heterogeneity. Figure 8(c) describes implementing a XML schema to ontology or semantic model. In the ontology document, properties (has_name, has_firstName, has_lastName, has_street, has_city, has_zip, has_state, and has_country) have attached to the elements of the domain (Address) to help discover a value of an element based on semantic matching.

```
<complexType name = “Address”
sawsdl:modelReference = “Ontology#Address”
sawsdl:liftingschemaMapping =
“http://www.oum.edu.my/schemaMapping”>
<sequence>
<complexType name = “name”>
<sequence>
<element name = “firstName” type = “string”/>
<element name = “lastName” type = “string”/>
<sequence/>
<complexType/>
<element name = “street” type = “string”/>
<element name = “city” type = “string”/>
<element name = “zipCode” type = “string”/>
<element name = “state” type = “string”/>
<element name = “country” type = “string”/>
</sequence>
</ complexType>
```

Fig8 (a): WSDL complex type element

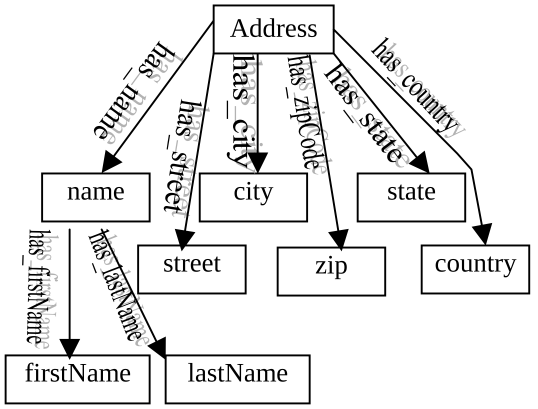

Fig8 (b): OWL Ontology

```
<owl: Class rdf: ID = “Address”>
      <owl: Class rdf: ID = “name”>
           <owl: subClassOf rdf: resource = “#Address”>
           <owl: has_firstName datatype = “string”>
                {getFirstName(firstName)}
           </owl: has_firstName>
           <owl: has_lastName datatype = “string”>
                {getLastName(lastName)}
           </owl: has_lastName>
             </owl: subClassOf>
      </owl: Class>
      <owl: has_street datatype = “string”>
           {getStreet(street)}
      </owl: has_street>
      <owl: has_city datatype = “string”>
           {getCity(city)}
      </owl: has_city>
      <owl: has_zip datatype = “string”>
           {getZip(zip)}
        </owl: has_zip>
        <owl: has_state datatype = “string”>
           {getState(state)}
        </owl: has_state>
        <owl: has_country datatype = “string”>
           {getCountry(country)}
        </owl: has_country>
</owl: Class>
```

Fig8 (c): Lifting Schema Mapping

5. EVALUATION

The purpose of this paper can be considered to explore knowledge and understanding in the domain of Web service to facilitate interoperation between Web services. Message level heterogeneity is one of the cases web services face in times of interoperation between them. As shown in figure 5, WS1 (web service one) sends message in one format to WS2 (web service two). WS2 receives the message in another format. Because of format variations, direct data transmission faces problems which require solutions through data mediation. The main task of data mediation is to transform source schema (data structure) to target schema. The transformation is performed two ways: 1) Manual, and 2) Semi-automatic. To raise the level of automation, we need machine translation from one format to another. Generic approach of schema transformation is necessary. Without generic approach, previously built schemas cannot be reused if service requirements and message patterns changes. Therefore, previous effort and investment cannot be utilized. In the generic approach, a conceptual model is introduced to mediate between web services. The conceptual model known as ontology is a dictionary like thing that holds common terms used by the web services. Semantic annotations are attached to the terms (elements). The ontology is implemented using OWL (Web Ontology Language). In times of interoperation between two web services, matching of elements from source schema is searched against elements of target schema. Matching can be incorrect if relations between the terms are not resolved through consultation with the ontology, in which terms are related by semantic annotations. If matching between elements is not found in the ontology, a super-class element can be used as a matching for a subclass element. As shown in figure 8 (b), to convert from {firstName, lastName} to {name}, we need to search from subclass to super class. To map from {name} to {firstName, lastName}, we need to search from super-class to subclass. Schemas are tree like structure. Every element of source schema is required to convert to a corresponding element of the target schema. Therefore, we need to visit all elements of source schema in depth-first manner. On the other hand, elements of ontology are visited only if matching problem occurs and visiting is limited until the matched element is found. There are no general rules available to build and search ontology tree. Therefore, it is necessary and important to shed light and conduct research on building, searching, and maintaining ontology, which is used to define concepts and to give structure to data or messages.

6. CONCLUSION

The purpose of this paper is to explore knowledge and understanding in the domain of Web service to facilitate interoperation between Web services. Message level conflict or heterogeneity creates problems in the interoperation of web services. Because of format variations in messages from source to destination or target, direct data transmission faces problems. Data heterogeneity or message level conflict in web services is not a new problem.

For nearly a decade, vigorous industrial and academic research on interoperation between web services has been going on resulting in fundamental and complex tools as well as technologies including WSDL, SAWSDL, SOAP, OWL, XML, XQuery, Xpath, and XSLT. The solution to the message level conflict is the data mediation. The main task of data mediation is to transform source schema (data structure) to target schema Extensive research has also been done to advance the mediation from manual to semi-automation. Because of complex data structure and involvement of many complex technologies, achieving complete automation of data mediation is a far distance. To raise the level of automation, we need machine translation from one format to another. Automatic data mediation requires a conceptual model known as ontology. The ontology acts as a special kind of dictionary that holds common terms used by the web services.

Proper matching between elements of source schema (data structure) and target schema depends on the semantics of the words. The similar the match, the more perfect is the level of automatic data mediation. When direct match between elements of source schema and target schema fails, ontology is searched to find a semantic matching. This paper describes fundamental architecture, tools, and technologies involving Web service. It focuses on web services in the context of data mediation, which is required to remove data heterogeneity or message level conflict between web services. This paper proposes how to improve the level of automation of data mediation through semantic matching of elements of source and target messages. This paper emphasizes that it is necessary and important to shed light and conduct research on building, searching, and maintaining ontology, which is used to define concepts and to help find semantic match between two message elements.

REFERENCES

1. I. Singh, S. Bryndon, G. Murry, V. Ramachandran, T. Violleau, B. Stearns: Designing Web Services with the J2EE 1.4 Platform. Addison-Wesley, 2004.
2. C. Wu and E. Chang: An Analysis of Web Service Mediation Architecture and Pattern in Synapse. May 2007, IEEE Computer Society, Niagara Falls, Canada
3. http://www.w3.org/2002/ws/sawsdl/. Accessed on 2009/02/26.
4. http://www.epa.gov/webservices/glossary.htm. Accessed on 2009/02/26.
5. http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s02.html. Accessed on 2009/02/27.
6. P. Kunfermann and C. Drumm: Lifting XML Schemas to Ontologies – The Concept Finder Algorithm. Proceedings of 1st International Workshop on Mediation in Semantic Web Services (MEDIATE 2005), December 12, 2005, Amssterdam, Netherland.
7. M. Nagarajan, K. Verma, Amit P Sheth, and John A Miller: Ontology Driven Data Mediation in Web Services. International Journal of Web Services research, Volume 4, Issue 4, 2007.
8. M. Ferdinand, C. Zirpins, and D. Trastour: Lifting XML Schema to OWL. Proceedings of the 4th International Conference on Web Engineering, July 26, 2004, Munich, Germany.
9. H. Bohring and Soren Auer: Mapping XML to OWL Ontologies. Leipziger Informatik – Tage, 2005, Leipzeg, Germany. http://www.informatik.uni-leipzig.de/~auer/publication/xml2owl.pdf. Accessed on 2009/03/08.
10. D. Martin, M. Paolucci, and M. Wagner: Toward Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective. Proceedings of the 4th European Semantic Web Conference (ESWC), 2007, Innsbruck, Austia.