ENHANCED LAYER BASED MODEL IN SUPPORT OF WEB SERVICES INTEROPERABILITY

SHAHZADA ZEESHAN WAHEED1, ABDUL WADOOD2, DR. FAROOQ AZAM3

1DCE, College of EME, (NUST), H-12, Islamabad, Pakistan
szw.shani@gmail.com
2DCE, College of EME (NUST), H-12, Islamabad, Pakistan
aww1615@gmail.com
3DCE, College of EME, (NUST), H-12, Islamabad, Pakistan
farooque.azam@gmail.com

ABSTRACT. Web services are the emerging technology used to perform complex tasks. Web services are available over the internet and different application can request a specific service to perform the required tasks. As web services are used by different type of applications having different models and protocols, maintaining the interoperability are very important. In this paper we consider the different techniques, standards and models used to maintain interoperability, i.e. WSDL-S, SOAP engine and UDDI (Universal Description Discovery and Integration)[5]. As web services are used commercially then main issue is the security. Integration layer Model is available that is used in support of interoperability in web services [1]. In this model, layers are divided in to low level (with more interaction) to high level (with less interaction). Main purpose of this paper is to propose the new enhanced based model. This model is based upon the layer architecture, to support message passing between services having different syntax, semantic and underlying platforms.

Keywords: UDDI, SOAP, WSDL-S, XML

1. Introduction. Web services are the application components that are available for open use. It provides the services that can be used by different applications. Services do not need any other information because these are self explanatory. If we see globally there are different protocols and techniques are used, so how these services are searched by the applications? Simple answer is UDDI. Universal Description, Discovery and Integration are a standard for services to be registered. It is a standard for creating, discovery and distribution of web services. Web services communicate through open protocols.

Web services are self descriptive. It describes the data format and protocol used to coordinate with this web service. Understand the architecture of the web service in the form of layer is the best possible approach. Different layers are divided considering the level of interaction between systems.

Main purpose of the web services is to provide smooth communication between heterogeneous systems. This can’t be achieved simply. Heterogeneities can occur at any layer of communication. For example at the transport layer different systems must agree upon the underlying physical transport mechanism. We can’t expect Java Messaging Service to be invocated on non java platform.

Our main focus in this paper is to identify the best possible approach to add loose coupling features between systems.
Following are the main points focused in this paper:

1. Using Layered approach adding an extensive feature in the web services to provide ability of heterogeneous system to interact with the web services.
2. Instead of using the same protocol for passing messages and using services making web service capable to interact with any heterogeneous system having different data model and different message syntax.
3. Adding loose coupling capability with the services consumer and services.

In this approach we propose model that helps in passing messages and using services of the service provider without conforming to the protocol and syntax required by the service. This is will increase the loose coupling and late binding allows the systems to interact with web services without conforming to the standards required by this specific service.

2. Related Work. Interoperability is the main concept in Web services. Main issue is the coordination or agreement on the one possible protocol that will be used in communication. Web service is the evolutionary technique in Services Oriented Architecture. Many researches are done in respect to message passing and using systems in between heterogeneous systems. Semantic Interoperability approach is used in message passing between different systems that are using same protocol to pass message [2]. In this technique one system can use the services if it agrees to the semantic used and provided to this specific web service. The developer of the system must understand the syntax and semantic provided by that service. Layered approach is the very good approach in understanding the web services at the different abstraction levels [1]. In this model layers are divided by considering the coordination made between integrated systems and the services. M Nezbad describes in his paper that we can easily understand the error by understanding this concept to the forms of layers. Audition Framework used for the web services interoperability testing was one step ahead to understand and test the services interoperability [3]. It uses Enterprise Application Integrations (EAI), main purpose of this technology to make different applications running to the different application domains to interact with each other. In this approach applications are running on different platforms but there applications are required to conform to the same communication model to interact with each other.

3. Problem and Solution. We can understand the layers, how these layers are divided in support of heterogeneous message passing between applications using different data models. We propose a new intermediate layer and all the communication is done through this layer. We will explain the purpose of this new layer using following diagram.

![Diagram of Enhanced Layered Model of Web services.](image)

3.1. Intermediate Layer in Newly Proposed Model. Main Enhancement of this layered model is the addition of new layer between service and service consumer. As shown in fig 1, this layer is responsible for the late binding. This layer can be understand by subdivided this layer as follows:
As Fig. 2 shows, the intermediate layer is divided into three sub-parts to understand its functionality. First part is the understanding the protocol. This is responsible for understanding the protocol and conventions used by the system which is going to interact with this service. System that is requesting the service, need not to be in the format that is used by this specific service. This is the responsibility of the second part of the intermediate layer that is “Dynamic Binding”. Dynamic binding ensures in the runtime that the services that are requested are conforming to the requesting protocol. This will enable the heterogeneous systems to interact with each other easily. Third one the coordination part, it enables the error free message passing between different systems and making enable to use services easily. This is increase dynamicity in the web services.

We can understand the problem considering the C# example of the web services. In this example we have created a sample web service that includes the hello world method. This method accepts the int argument and displays it.

```csharp
namespace WebService1
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [ToolboxItem(false)]
    {
        [WebMethod]
        public string HelloWorld(int id)
        {
            return "Hi, your ID is: " + id.ToString();
        }
    }
}
```

Fig. 3 Creating Simple Web Service
What will happen if a system calls the services of this web service without conforming the syntax? When we call this HelloWorld method to test this web service, it runs perfect when we provide it an integer argument. Our point is what will happen when we call this service providing it variable of any other type like string etc.

It generated an exception and operation was not completed. We can implement exception handling to avoid this message and do something else in this situation. But point is to understand the schema that is required and then convert the request according to the required Schema.

The following is a sample SOAP request and response.

```
POST /Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/HelloWorld"

<?xml version="1.0" encoding="utf-8"?>
  <soap:Body>
    <HelloWorld xmlns="http://tempuri.org/">
      <id>int</id>
    </HelloWorld>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
  <soap:Body>
    <HelloWorldResponse xmlns="http://tempuri.org/">
      <HelloWorldResult>string</HelloWorldResult>
    </HelloWorldResponse>
  </soap:Body>
</soap:Envelope>
```

This XML shows the required parameters and their type (highlighted in above XML code). There is an external mechanism that can handle these heterogeneous issues. In the conceptual model (mentions above) adding a new layer between messaging layer allows to analyze the required interface and then act as a middleware to map the provided interface to the required one. Proposed intermediate layer can be explained as middle layer between two services. Message first arrives in middle layer; this layer is capable for analyzing the message formats. Once message is understands then this layer converts this message into the format that receiving layer is expecting.

- Acting as a middleware for messaging
- Receive messages from the both web services
- Capable of manipulating the received messages
Has knowledge of syntax and semantic used at the receiving end web service
- Converts the message into to format expected at the receiving end
- Transmit the manipulated message to its final destination.

Above figure shows the main structure of the intermediate layer. Here we will discuss each part in detail.

This is divided into two parts. We can understand it in sequence of steps.

**Registering web services:**
1. Developers develop a web service and then send it to register in UDDI.
2. There is an intermediate step before registration of the web service. Semantic of this web services is snatched and analyzed.
3. This semantic is stored into semantic database.
4. Finally web service is registered into UDDI and it updates its WS Proxy, making it available for other clients or web services.

**Using web service:** When a web service wants to used the services of other web servicers, following steps are involved in this process.

1. Intermediate layer understand and analyze the semantic of the web service its and the other web service which is going to be used.
2. Based on the analysis, intermediate layer generates the matching syntax for providing heterogeneous web services to communicate with each other.
3. Ws proxy involves in discovering and communication between different services.

There are two data stores that are involved in intermediate layer.

1. Registered Services: This is responsible for storing the registered web services and making that those services available for use.
2. Semantic DB: This is responsible for storing the different semantics of the web services. Analyzer works on this data store, it matches the semantic of the web service with the data stored in it. And finally generates the suitable matching syntax, enabling the heterogeneous web services to communicate correctly.

We can understand the above concept by using a simple concept as: Web services of Invoice Payment consist of interface like payInvoice. This interface would be used for using its services; we can call it web service interface. Different rules that would be used for operating of Invoice Payment are the business rules of this specific domain. Specification may contain the meta-specification that helps in identification of web service.

**3.5. Web Services Approaches for Interoperability.** There are two main approaches that are available proving interoperability specifications: one is the WS family and the last but not the least is the ebXML [7]. These are the
This paper includes our research in web service specific standards and using different issues in the web services certification. Depending on the analyzer, communicating web services to communicate with other systems requesting the service and web services [15].

We based our research on a new standard specification approach. The intermediate type converter approach help to receive the request from any system and then covert its request to the technically matching with web service. Any system having any new or updated specification can now connect with web services without worrying about the interface that would be used for interaction. ebXML is responsible for document exchanging. Type converter can be based upon the ebXML specification; it manages the sequences of the message passing. Developers can use the combination of the above two approaches in defining the interoperability approaches between different services. This concept can be treated similar to WS-Trust, and WS-Federation combination, that provide features such as establishing trust and service federation on top of WS-Security [8].

Service provider defines the specification that is used in using the services provided by this specific web service. Heterogeneous systems can communicate and use the services if they agree upon the specification provided the developer of specific web service. But using the intermediate layer and converter enables any type of the heterogeneous system to communicate and use the services of specific system without agreed upon the specification. It leaves all the coordination tasks on the intermediate layer. This layer enables different systems to communicate and use web services effectively proving security and reliability.

4. Future Work. In intermediate layer we proposed the new analyzer component. It analyzes the web service from its XML description and stores it into semantic DB. This component is responsible for accessing the semantics of different web service and the generating the matching semantic. This is not too simple as different type of heterogeneities can occur that should be countered using this layer. Continuing this research it enables us developing the intelligent analyzer that access the web services and then automatically generate the matching syntax for effective communication between heterogeneous web services.

5. Conclusion. This paper includes our research in web service interoperability. Main issue is the heterogeneous web services communication [11]. In case any web services are not following proper standards for communication, they can’t communicate with each other. We based our research on layered based model proposed by Hamid and Fabio. They proposed the layered model for web services and we enhanced that model adding an intermediate layer. This intermediate is responsible for making the heterogeneous web services to communicate with each other. Heterogeneities can occur if web services are not conforming to the specific standards and using different syntax and semantics conforming to other standard.

We can summarize our discussion as main issue in the web services communication is heterogeneities. Our solution provides an adequate way to resolve this issue. Implementation of this solution will make it possible for heterogeneous web services to communicate and use services correctly. All the communication gaps are resolved in the powerful intermediate layer. Depending on the analyzer, identifying the syntax and semantics of different web services make it possible to automatically remove the differences and find an optimal way to make the different web services communicate with each other and use their services.

REFERENCES


