Using Web Services For Distributed Computing

Neeraj Saxena
University of Delhi, Delhi, E-Mail: neerajom@rediffmail.com

Anita Goel
University of Delhi, Delhi, agoel@dsc.du.ac.in

ABSTRACT
A web service facilitates integration of various applications and interoperability among distributed heterogeneous application and components. It is a standard representation for information resource that can be used by other programs, regardless of the development platform, middleware, operating system and hardware type. This paper discusses using the web services for distributed computing. We discuss the architecture of components of web services and the fundamental standards – SOAP, WSDL and UDDI that enable communication between web services. We illustrate the working of web service and binding of web service to an application, with the help of an example. The paper also highlights the security concerns associated with web services.

KEYWORDS
Web Services, XML, SOAP, WSDL, UDDI.

INTRODUCTION
In distributed application software, the functionality of the software is spread across different distributed computing nodes. The middleware component of the distributed systems facilitates interaction between heterogeneous computing applications within one organization or among more than one organization. However, integration of business processes among the organizations is dependent on the kind of development platform, the operating system used and the hardware type. Web service technology allows integration of application across different organizations and facilitates interoperability among distributed heterogeneous applications and components, independent of the development platform, middleware, operating system and hardware type.

Web service is application software that is accessible to other application software over the web. According to W3C[7], a web service is “A software application identified by a URI whose interface and bindings are capable of being defined, described and discovered as XML artifacts”. By defining web service, any remote user (consumer i.e. who wants to bind existing service and reuse it) can use this application by developing another software application which can invoke this existing web service. Web service supports direct interaction with other software agents using XML based messages exchanged via internet based protocols. Web service allows different applications from different sources to communicate with each other without expenditure of time and coding. Web services are not tied to any one operating system or programming language. It is inter-working between different organizations.

Web service exposes the internal operations of an application so that they can be invoked through the web. The creation and use of web service is described using service provider, service requestor and service registry. Web service has service providers that create the web service. Additionally, the service provider also service description for the web service and publishes the service description in a service registry. Service registry uses the description to segregate the different services and catalog them. Information about the connection to web service and interaction with web service must be accessible to interested parties [8]. A service requestor who wants to use a web service, queries the service registry. From the query result, the service requestor can view the service description and the location of the service provider. The service requestor then binds to the service by invoking it.

Web services integrate web applications using XML, SOAP, WSDL and UDDI standards. The XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available, and UDDI is used for listing what services are available.

Web services are both - simple and composite. A simple web service is internet-based application that does not rely on other web services to fulfill consumer request. For example, provide history information about cars. A composite service is defined as a combination of two or more web services that are integrated to offer a value-added service. An example of composite service is a car broker that out sources to car dealer, financing and insurance services to provide “complete” car sale solutions [3], to enhance the reusability and minimize the cost. The paper discusses the components required to define a web service and using (binding, consuming) an existing web service. The paper illustrates defining of a web service and consuming a web service with the help of an example. IBM Certified Enterprise Developer and Sun Certified Enterprise Architect for J2EE and Microsoft .NET are being used for defining and consuming a web service explained later in the paper.

We list some of the free available web services and their purpose. Security issues are also discussed in this paper.

COMPONENTS OF WEB SERVICE ARCHITECTURE
The architecture of web service includes the XML, HTTP, SOAP, WSDL and UDDI [7]. XML is common grammar of web. XML is used to write WSDL and SOAP address of the web service. SOAP is a protocol that acts upon HTTP for
generating XML schema of information which is to be exchanged in distributed environment. WSDL is a document written in XML to describe the web service reference and the available methods. UDDI is a repository of web services. All these components are explained as follows:

XML (Extended Markup Language): It is a common grammar used to define a public interface. It improves the functionality of the web by identifying information in a more accurate, flexible and adaptable way. All requests are encoded in XML so that messages can be understood at any end. XML is sent via HTTP POST.

HTTP (Hypertext Transfer Protocol): HTTP is a request/response protocol between clients and servers, used to transfer information on the World Wide Web. HTTP is used for transporting message between applications.

SOAP (Simple Object Access Protocol): This protocol is used for exchanging information in a distributed environment. It organizes information using XML in a structured and typed manner so that it can be exchanged between peers. SOAP message defines how clients can invoke a remote procedure, and how service can reply by sending another SOAP message back to the caller.

WSDL (Web Service Definition Language): WSDL specification is a XML document that describes web service, service interface, service reference (location), methods and data types available in web service. WSDL includes a collection of different interactions paradigms along with the ability to combine operations.

UDDI (Universal Description Discovery and Integration): It is a naming and directory service which defines data structures and APIs for publishing service description in the registry and for querying the registry to look for published descriptions, respectively. UDDI specification allows clients to query the registry and obtain references to services of interest. Four main entities in the registry are:

- Business entity: It describes the organization that provides the web service.
- Business service: This describes group of related web services offered by a business entity.
- Binding template: This describe the technical information necessary to use a particular web service. Essentially, it defines the address at which the web service is available and a set of detailed information.
- Technical model (tModel): It represents a WSDL service interface, interaction protocols and semantics of operation.

DEFINING A WEB SERVICE

Web services are software applications being developed in .NET or J2EE for distributed environment. Web services are concerned with four tasks [8]:

1. **Service description** provides detailed explanation about the web service functionality, how to access web service, address of web service, methods and data types defined in web service. A **Service implementation** includes designing a framework so that web service can receive a request and can send the response to requestor.
2. **Service publishing, discovery and binding** is required to expose and publish the implemented web service so that the web service could be consumed by different users in the distributed environment. d) **Service invocation and execution** relates to invoking of the web service and its execution by the service requestor. Here, we briefly explain the four tasks:-

**Service Description:** The web service description includes common base language, interfaces, business protocols, properties and semantics and vertical standards. XML is used for common base language because it has syntax flexible enough to enable the definition of service description. The web service interface definition is provided using WSDL. Web service often offers a number of operations that clients must invoke in certain order to achieve its request. Such exchanges between clients and web services are called conversation. Service provider want to impose rules that govern the conversations, such set of rules are called business protocol. Web services provide information to facilitate binding in autonomous and loosely-coupled setting, this may include non-functional properties such as the cost or quality of a service, and this is information that is crucial for using the service.

**Service implementation:** This includes structuring data and operation inside of an XML document that complies with the SOAP specification. Once implementation is complete, client (consumer) sends message to the service as an XML document and the service sends an XML document back to the client as response.

**Service publishing, Discovering and Binding:** Once a web service has been implemented, it must be published some where that allows interested parties to find it. Information about how a client would connect to a web service and interact with it must also be explained. Only after publishing of a web service into registry (UDDI), web service can be discovered and bind. There exists a Service-Requester who requests some service with the help of HTTP. The service requester finds the service in service registry. Binding of service provider and service-requester is achieved with help of an interface (application) based on standards (protocols) SOAP, WSDL and UDDI.

**Service Invocation and Execution:** Web service invocation and execution is accomplished by implementing a web service listener i.e. another software application which can gain access to a web service. In order to implement this, a system needs to understand SOAP message and generate SOAP response as per the WSDL of web service. When a web service request is received, the SOAP listener validates a SOAP message as defined in WSDL file. The SOAP listener then invokes the
corresponding web service and finally, business logic is invoked. The result of the business logic is transformed into a SOAP response and returned to web service caller.

WORKING WITH WEB SERVICES
For working with web services there needs to be a web service provider (WS-Provider) who provides a web service, web service consumer (WS-Consumer) i.e. user who wants to use a web service, and, a mechanism to search for a web service. The web service provider provides web services, the web service consumer searches for the available web services and consumes it. In this section, we discuss about (a) WS-Provider, (b) Searching for a Web Service and (c) WS-Consumer.

WS-Provider: A web service provider creates an application that implements a particular service to be provided. WS-Provider writes WSDL document to describe the web service in UDDI registry. The web service provider registers the web service in a UDDI registry with the help of a DISCO document (an example explained further). Provider of a web service may bind already registered services using SOAP address described in DISCO document.

IBM Certified Enterprise Developer and Sun Certified Enterprise Architect for J2EE and Microsoft .NET are being used to develop a web service and to consume a web service. WSDL and Discovery documents are generated by itself in Internet browser if the application is created as a web service.

Searching for Web Service: A web service can be searched using a search engine (for example google.com), or, by using the service standard, namely, UDDI (Universal Description Discovery and Integration). UDDI is a registry that provides a list of all available web services in different categories (stock and shares, hotel booking etc.).

WS-Consumer: The web service consumer uses the UDDI browser [8] to find the web service to be consumed or bind. The UDDI browser gives the following information about the web service-

- Discovery of web service provides information about exact URL of web service, provider of web service, SOAP address, and, terms and condition for using the service, and
- WSDL document provides information about the purpose of web service, functionality of service, data types, parameter required to call methods, and URI reference as namespace.

The web service consumer checks the information displayed by UDDI browser and then binds to the desired web service.

AN EXAMPLE OF .NET WEB SERVICE
Here, we present an example to provide and consume a web service. We show how to create a web service, the WSDL document for the created web service, discovery (DISCO) document of web service, and consuming the described web service. We create a web service named WsCalc.asmx in .NET framework. The web service WsCalc.asmx accepts any two integers and displays their sum.

Web Service creation- WS-Provider creates a web service named WsCalc.asmx, using the steps shown in Table 1. In step 1, the ASP.NET web service framework is opened. Steps 2 and 3 are for specifying a URL and name of web service, respectively. Step 4 describes the method called CalAdd which accepts two integer values and returns sum of two integers in XML format.

<table>
<thead>
<tr>
<th>Table 1: Web service creation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Start Visual Studio.NET and Select New&gt; Project&gt; Visual C# Project&gt; ASP.NET Web Service</td>
</tr>
<tr>
<td>2. Write <a href="http://localhost/NsCalculator">http://localhost/NsCalculator</a> on location field.</td>
</tr>
<tr>
<td>3. Rename Service1.asmx to WsCalc.asmx</td>
</tr>
<tr>
<td>4. Create a web method called CalAdd as follows</td>
</tr>
</tbody>
</table>

```csharp
[WebMethod (Description = "This Web Service would accept two integer values and return the sum of the two integer received.")]
public int CalAdd(int num1, int num2)
{
    return (num1+num2);
}
```

WSDL Document of WsCalc – WSDL document of the web service discussed in Table 1 will be generated and stored at URL- http://localhost/ NsCalculator/WsCalc.asmx?WSDL. Table2 shows the WSDL document of WsCalc.asmx. The lines marked bold in Table2 gives the details of method and binding information of web service WsCalc.asmx.

<table>
<thead>
<tr>
<th>Table 2: WSDL Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;?xml version=&quot;1.0&quot; encoding=&quot;utf-8&quot; ?&gt;</td>
</tr>
</tbody>
</table>

```
<wsdl:types>
    <s:element name="CalAdd" type="s:int" maxOccurs="1"/>
</wsdl:types>
```
DISCO (Discovery) Document of WsCalc: The Discovery file of the web service WsCalc will be generated and stored at URL http://localhost/NsCalculator/WsCalc.asmx?DISCO. Table 3 shows DISCO file of WsCalc.asmx. The lines marked in bold in Table 3 gives SOAP address which is the key requirement for exchanging the request and response of the web service.
Using Web Services For Distributed Computing

**Consuming WsCalc Web Service:** To consume the web service *WsCalc*, consumer will create the application as explained in Table 4. In Table 4, Step 2 tells about the client location, and steps 3, 4 are for binding reference of above web service *WsCalc*. Step 5 and 6 is to design web form for invoking method *CalAdd* of *WsCalc* web service. Table 4 also shows lines of code for invoking method *CalAdd* of web service *WsCalc*.

<table>
<thead>
<tr>
<th>Table 4: Web service invoking code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Open the Microsoft Visual Studio.NET</td>
</tr>
<tr>
<td>3. Add reference of above web service: Right click on References. Click Add web Reference. Enter <a href="http://localhost/NsCalculator/WsCalc.asmx">http://localhost/NsCalculator/WsCalc.asmx</a> in URL field. Click go</td>
</tr>
<tr>
<td>4. Enter CalcWebServices to Web reference name field and click add reference</td>
</tr>
<tr>
<td>5. In the design view of the WebForm1.aspx, Add two Text boxes, one Label and one Button.</td>
</tr>
<tr>
<td>6. Double click on the Button and make sure the Click function of button is as follows.</td>
</tr>
</tbody>
</table>
```csharp
define protected void Button1_Click(object sender, EventArgs e) |
{ CalcWebServices.WSCalculator mycal = |
    new CalcWebServices.WSCalculator(); |
    Label1.Text = Convert.ToString(mycal.CalAdd(Convert.ToInt16(TextBox1.Text), Convert.ToInt16(TextBox2.Text))); } |
```
| Table 4: Web service invoking code |

Figure 2 shows display output on executing the web service WsCalc.asmx (Table1) at URL- http://localhost/NsCalculator/WsCalc.asmx. The method *CalAdd* of web service *WsCalc* is executed on clicking CalAdd and output of method *CalAdd* of web service *WsCalc* is displayed.

The run time snap shot of this web service is shown in Figure 3. To test the functionality of the web service, enter 10 and 16 in num1 and num2 respectively. When we click on invoke button we get the result in XML format i.e. 26, as shown below. Now on running web service invoking code as shown in...
SECURITY AND RELIABILITY ASPECT
Using a web service requires exchange of number of messages among various parties over networks. As a result, security is a key issue to protect the privacy and safety of data. Some mechanisms for the security of web services are as follows [2]-

- **Message Integrity:** To protect message from tampering, ‘WS-Security’ defines a SOAP header that allows digital signature to be applied to the message content.
- **Trust Management:** To facilitate secure communication across trust boundaries, WS-Trust provides a web service based protocol for brokering trust between two or more parties.
- **Reliability:** Architecture of web service does not ensure end-to-end transmission through a reliable path as web service relies on multiple parties. But, ordered delivery of messages sent by many parties can be sequenced with the help of ‘WS-reliable messaging’. It defines a sequence of SOAP headers to correlate multiple messages in a sequence.
- **Acknowledge delivery:** WS-reliable messaging provides a transport-neutral acknowledgement system that allows sender and receiver to find, which messages have and have not been successfully transferred.
- **Transmission Control:** WS-transmission control specification provides mechanism for advertising when a given service is and is not available.

GENERIC CLASSES OF WEB SERVICES
The available web services are categorized based on their functionality. Some categories of web services are as follows-

- e-Governance Service(Govt. Rules, policy etc)
- e-Commerce Service(Banking, Invoicing etc)
- Road-mapping Service(Microsoft Map Point)
- e-Media Service(news paper, Journals etc)
- Travel and tourism Service( India tourism etc)
- Environmental Services(Weather)
- Educational Services(du.ac.in etc)
- Microsoft Services(downloads, updates)
- Bio-Medical Services
- Postal-code service

FREE WEB SERVICES
Web services are available free of cost and as well as on payment term and conditions. Here we list some available free Web services-

- 2ip.com: Free web space, hosting, website, email and domain name services.
- freepolls.com: Web site polls and professional banner.
- webservicesfree.com: Employment and Internet research service, article publishing service.
- conceptdraw.com/visio: Diagramming software.
- aws.amazon.com: Access several Amazon Web Services for free.
- afterzed.com/free/webservices: Links to sites that offer many free services.
- trynt.com: Manufacturer of fine free web services and APIs.
- gartner.com: Affiliates to offer Amazon.com content and features.

CONCLUSION
Web service technology allows integration of application across different organizations. It facilitates operations between heterogeneous applications and components. Web services provide interaction independent of the development platform, middleware, operating system and hardware type. UDDI, XML, SOAP, WSDL are main components of web services.

REFERENCES