Developing a Cross-Platform Web Service using SOAP

Aye Min Than, Ei Ei Han
University of Computer Studies, Yangon
thadar2006@gmail.com

Abstract

In this paper, the proposed system can give service over different platform. SOAP (Simple Object Access Protocol) is an application of the XML (Extensible Mark-up Language) specification. It relies heavily on XML standards, like XML Schema and XML Namespaces for its definition and function. XML messaging is where applications exchange information using XML documents. It provides a flexible way for applications to communicate, and forms the basis of SOAP. All SOAP messages are encoded using XML. The proposed system demonstrated the comparison of the implementation of SOAP web services under different platforms (PHP and ASP), and how to request data from web server which is implemented on Microsoft Platform and show that requested data on the web application which is implemented with Open Source Platform.

.

Keywords: Web Services; SOAP (Simple Object Access Protocol)

1. Introduction

The execution of business processes across the internet is the core of e-business and e-commerce. These business processes are not static, but dynamically evolving. Many services have to interoperate seamlessly, such as ordering, billing, calculation services etc. In addition, the set of services is not fixed, because business process changes and extensions require the integration of additional services. The services are scattered around different enterprises and implemented using different platforms, programming languages etc. Due to the fact, that there is no centralized organization, the services are evolving independently. In summary, the execution of business processes means not only the distributed execution of business processes, but also the integration of an evolving set of heterogeneous services.

There are several types of web services available in these days. Among those, SOAP web services provides a flexible way for application to communicate and it is designed to be a new protocol for the decentralized, distributed environment, which utilizes the power of the Internet and XML to pass typed information between nodes [1].

SOAP stands for Simple Object Access Protocol. It is a communication protocol for communicating between applications. SOAP is a format for sending messages and designed to communicate via Internet. It is platform independent and language independent. SOAP is based on XML. It is simple and extensible. SOAP allows you to get around firewalls and it will be developed as a W3C standard [8]. It relies heavily on XML standards like XML Schema and XML Namespaces for its definition and function [8]. XML messaging is where applications exchange information using XML documents. It provides a flexible way for applications to communicate, and forms the basis of SOAP [8].

All SOAP messages are encoded using XML. A SOAP application should include the proper SOAP namespace on all elements and attributes defined by SOAP in messages that it generates. A SOAP application must be able to process SOAP namespaces in messages that it receives. It must discard messages that have incorrect namespaces and it may process SOAP messages without SOAP namespaces as though they had the correct SOAP namespaces [8].

The proposed system demonstrated the comparison of the implementation of SOAP web services under different platforms (PHP and ASP), and how to request data from web server which is implemented on Microsoft Platform and show that requested data on the web application which is implemented with Open Source Platform.

This paper is organized as follow: Section 2 discusses related work about SOAP Web Services. Section 3 discusses about Web Services and Section 4 mentions SOAP. The overview of the system design is presented in Section 5. Finally, implementations environment are presented in Section 6 and Section 7 is described as conclusion.

2. Related Work

The goal to integrate evolving sets of heterogeneous services has been already addressed by many approaches, which converge into the concept of middleware. There are different middleware technologies such as object-oriented middleware technologies and component-oriented middleware services, example CORBA [OMG] and DCOM. Those do not provide different implementations by different vendors and also require a large infrastructure, creating huge efforts in cost and time for their introduction. To avoid such problems, web services are a rising phenomenon in the network service world, and SOAP is the protocol of Web Services. One of SOAP's benefits is that, being XML, it is more human-readable than binary protocols and therefore easier to debug. Another often-mentioned benefit is the use of the ubiquitous HTTP as a tunneling protocol, which allows SOAP messages to penetrate firewalls. Furthermore, SOAP is a new lightweight protocol for the decentralized, distributed environment that is platform independent, transport independent, and operating system independent, all because it is built using time testing systems like the HTTP protocol and text mark-up in XML. Therefore, SOAP web service protocol is chosen for the proposed system.

3. Web Services

A web service is a software application whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web Service supports direct interactions with other software agents using XML-based messages exchanged via internet-based protocols [9]. Put it in a simple way, a Web Service is something one can call over the web form a program. Many organizations and companies (such as Microsoft, IBM, Sun and etc) are specifying the interoperability interfaces that enable software vendors to deliver services in the Internet's open, standards based environment.

For the end user it may not make any difference: "gets an input and displays the result". However, for developers/ programmers, Web Services makes their life easy. The main point is that platform independency as well as language independency can be achieved through Web Services. For example, instead of writing COM object if one exposes one's methods through Web Services using ASP.NET on Windows, it can access those methods in Java on Sun Solaris/ Linux machine, as shown in Figure 1. Because Web Services works on XML technologies, it makes developers life simple.

Web Services use XML to provide an application implementation of distributed application services and information based on multi-platforms and multi-languages. It means that applications and people can use the data and services of an application running on a remote computer regardless of differences between the application server technology running on the remote and local machines [9].

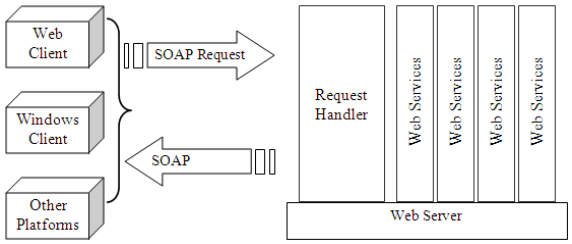

Figure 1: Typical Web Service

Web Services make it possible for diverse applications to discover each other and exchange data seamlessly via the Internet. While it is perfectly feasible for a person to use a browser to take advantage of a Web Service running on a remote computer, the real potential of Web Services lies in its ability to allow applications to work with other applications without the need for any human intervention.

Unlike other technologies, such as Sun's Enterprise JavaBeans, and Microsoft's DCOM, Web Services are platform and language independent. They also do not rely on complex language translations like COBRA. Web Services are independent because the messages sent between applications are in an XML-based protocol (called the Simple Object Access Protocol or SOAP). This allows applications written on one application server platform, such as ASP.NET, to be used by applications written in other platforms, such as Cold Fusion, PHP or Java [9].

4. Simple Object Access Protocol (SOAP)

SOAP's place in the web services technology stack is as a standardized packaging protocol for the messages shared by the applications. The specification defines nothing more than a simple XML-based envelope for the information being transferred, and a set of rules for translating application and platform-specific data types into XML representations. SOAP's design makes it suitable for a wide variety of application

messaging and integration patterns. This, for the most part, contributes to its growing popularity [5].

Once a WSDL description of a web service has been created, a service consumer must be able to locate it in order to be able to use it. This is known as discovery. In particular, Universal Description, Discovery, and Integration (UDDI) project and the new Web Services Inspection Language will be discussed later on. WSDL provides a service consumer with all the information they need to interact with a service provider. But how can a consumer learn of which services to use? The UDDI project is an industry effort to define a searchable registry of services and their descriptions so that consumers can automatically find the services they need. UDDI has two parts: a registry of all a web service's metadata (including a pointer to the WSDL description of a service), and a set of WSDL port type definitions for manipulating and searching that registry. The latest UDDI specification is Version 2.0. In this thesis, however, Version 1.0 is the point. Version 2.0 has not yet been widely implemented and there is very little support available for it [5].

UDDI is not the only option for service discovery. IBM and Microsoft have recently announced the Web Services Inspection Language (WS-Inspection), an XML-based language that provides an index of all the web services at a given web location.

4.1 SOAP Web Services

The basic principle of SOAP can be illustrated by a request response sequence, as it is used for remote procedure calls in a client-server relationship. The client can be implemented in any programming language and platform, the same applies to the server. The SOAP converter translates the programming language and platform dependent request of the client into a so-called SOAP request. This is a XML document. This document is sent to the receiver of the request, using standard internet protocols such as HTTP or SMTP facilitating the traversal of firewalls. On the server side, another SOAP converter decodes the XML-document representing the request into the format required by the server. After the server has processed the request, the converter encodes the result as an XML-document, a so-called SOAP-response that is sent back, converted, and delivered to the client [9]. SOAP Request/Response Sequence can be seen in Figure 2.

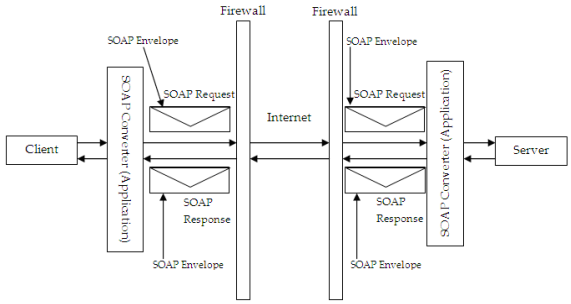

Figure 2: SOAP Request/Response Sequence

4.2 SOAP Messages

A SOAP message is an ordinary XML document containing the following elements:

- A required *Envelope element* that identifies the XML document as a SOAP message
- An optional *Header element* that contains header information
- A required *Body element* that contains call and response information
- An optional *Fault element* that provides information about errors that occurred while processing the message

A SOAP message consists of an envelope containing an optional header and a required body, as shown in Figure 3. The header contains blocks of information relevant to how the message is to be processed. This includes routing and delivery settings, authentication or authorization assertions, and transaction contexts. The body contains the actual message to be delivered and processed. Anything that can be expressed in XML syntax can go in the body of a message [4].

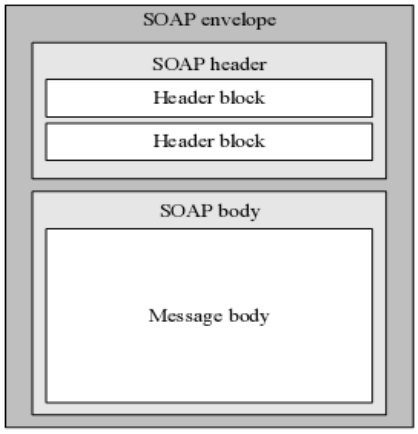

Figure 3: The SOAP message structure

5. System Framework

SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

PHP is simply a server-side scripting language. Server-side scripting allows sites to easily edit and update information, offer interactive features like forums and personalization, and track user traffic. ASP.NET allows users to program using Microsoft Internet Information Services (IIS). Microsoft software technologies enable a high level of software integration through the use of Web services, small, discrete, building-block applications that connect to each other as well as to other, larger applications over the Internet.

The main purpose of the development of this application is to emphasize the reliable of information exchange between heterogeneous web platform using SOAP web services. One web application will be developed under Microsoft .NET Framework and Microsoft IIS web server using SOAP web services. Another web application will be developed under Open Source Platform. Information shown in the Open Source Platform client web application are the information which is from the own web service and the information which is from different web service. If the requested information of the client is from the own web service, SOAP request will be requested to the web server for the requested information and then, response from web server will be responded to the client as XML document. Same transport method as above for the requested information of the client from the different web service. Because of same or different SOAP web services response will be send back to the client as uniform XML documents.

When user request data from the Client, the data will be requested to the Web Server with SOAP Request and the web server will get the necessary data from the Database and then give response to the Client with SOAP Response (XML) and the User will get response from the Client on User Interface.
The process will be same with the process from Microsoft .NET Platform. If the user requested exchange rate from the Client, the client will request data from Microsoft .NET Platform Web Server, instead of requesting to its own Web Server. The overview of the proposed system is shown in Figure 4.

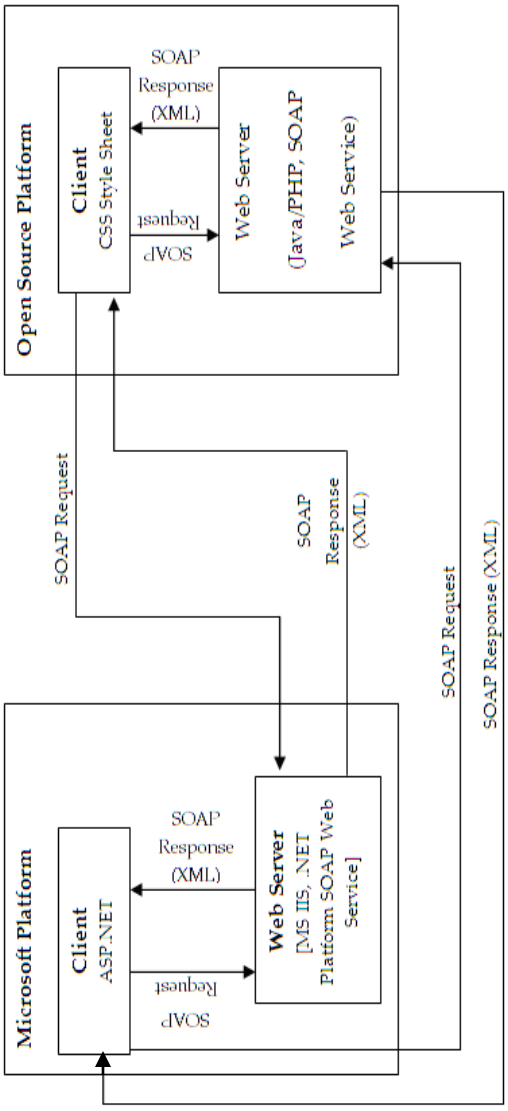

Figure 4: Overview of the System

6. System Implementation

SOAP based web services are designed with a common XML-based protocol. The goal is to allow for a machine readable document to be passed over any and/or multiple connection protocols to create a decentralized, distributed system. The proposed system demonstrated the comparison of the implementation of SOAP web services under different platforms (PHP and ASP), and how to request data from web server which is implemented on Microsoft Platform and show that requested data on the web application which is implemented with Open Source Platform. The web application which

developed using ASP.NET is the web site www.bookstore.com which is an online book shop. If a customer wants to get shipment for the book(s) he bought, the online book shop can give the customer how much the shipment service will cost by getting information from the web application which is developed using PHP web site www.shipping.com. The SOAP web service of www.shipping.com provides shipping rate information. The default currency of shipment price and the books prices are $. If someone wants to get other currency(£, ¥, Bat, S$, Ks), the SOAP web service of www.bookstore.com can support up to date exchange rate for US Dollar to £, ¥, Bat, S$, Ks. So, request site or client site and response site or server site are vice versa.

To develop SOAP Web Service in Microsoft Platform, Microsoft provides built-in service for SOAP in Microsoft Visual Studio .NET 2005. This system is developed with ASP.NET 2005, therefore, only SOAP Service namespace has to call as header file while we writing the program. In ASP.NET 2005, if the system wants to call the SOAP Service of other site, a web reference is added at the project. In this proposed system, web service of Shipping Company Web Site will be added as web reference. To get shippment charges, Microsoft platform must send three parameters(destination, transport type, itemweight) to opensource platform. The Shipping Web Site receives the sending three parameters from the Book Store Web site and replies the result from the calculate function depending on its. To develop SOAP Web Service with PHP, there is no built-in service for SOAP like Microsoft Platform. However, W3C Org provides the open source file like nusoap.php for SOAP Service. Like we call SOAP Service namespace in ASP.NET, we can call nusoap.php file as the header file. That nusoap.php file can be downloaded from the internet freely.

7. Conclusion

By using SOAP Web Services, data exchanging between different platforms can be processed easily with great performance. We can exchange any information on the network by using SOAP Web Services. Moreover, it can reduce that require a large infrastructure, creating huge efforts in cost and time for the different implementations by different vendors. Therefore, SOAP web services technology becomes more and more popular in these days. This technology still has a future, especially in business environments, where it has no real competitor and the focus is set on security. SOAP web services provide to implement World Standard Security and other related standards for big corporations and also be able to ensure suitable level of security and reliability.

REFERENCES

[1] Berners-Lee, (2002), WebServices: Program Integration across Application and Organization boundaries. http://www.w3.org/DesignIssues/WebServices.html.

[2] Doug Tidwell, Ken MacLeod, Pavel Kulchenko, Programming Web Services with SOAP, December 2001, O'Reilly & Associates, Incorporated.

[3] Kennard Scribner, Understanding SOAP : Simple Object Access Protocol, July 2000, Sams.

[4] Rebecca Riordan, SOAP Platform Web Service Development Using XML

[5] Steve Graham, Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI, May 2002, Sams.

[6] Steve Graham, Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI, May 2002, Sams.

[7] The World Wide Web Consortium http://www.w3g.org. Accessed on 15-08-2008

[8] Web Services architecture overview http://www-106.ibm.com/developerworks/webservices/library/w-ovr/. Accessed on 15-08-2008.

[9] Web Services architecture overview http://www-106.ibm.com/developerworks/webservices/library/w-ovr/. Accessed on 15-08-2008.