WEB SERVICES AND THEIR BUSINESS IMPLICATIONS

Bonn-Oh Kim, Seattle University, bkim@seattleu.edu
Ted Lee, University of Memphis, elee@memphis.edu
Yong Sok Choi, California State University, Bakersfield, ychoi2@csub.edu
Jinyoul Lee, State University of New York-Binghamton, jylee@binghamton.edu

ABSTRACT

Information technologies have been transforming and creating new business processes and subsequently organizational structures. In the 1990s, business process reengineering was enabled by client-server technology on local/wide area networks. Departmental walls and functional silos were removed or reduced due to these reengineering efforts. However, they were usually limited to one company. Web services available on the Internet in the 2000s provide an opportunity to broaden these reengineering opportunities to a chain of companies for inter-organizational business processes. This paper explains the web service technology and also discusses its business implications in terms of inter-organizational business process reengineering.

Keywords: Web services, business reengineering, web applications, XML

INTRODUCTION

In the 1990s, many companies in the USA redesigned their business processes and organizational structures in a way that corporate functional silos were replaced with cross-functional business processes spanning multiple departments in an organization (1, 3). The major enabling technology of this business reengineering process was client/server applications on the LAN (Local Area Networks) and WAN (Wide Area Networks).

Since the 2000s, the Internet has been playing a major role in business operations. Initially, the world-wide web on the Internet functioned primarily as a static publication tool and later has become an interactive site for simple business transactions as can be seen in many e-commerce retail sites. Recently, it is becoming widely recognized as a tool for integrating business applications across multiple organizations. Even though some technologies have been available for enterprise applications integration, it has been quite expensive and difficult to implement, e.g., EDI (Electronic Data Interchange).

The web service is a very promising technology enabling the enterprise applications integration across multiple organizations on the Internet. It is based on open standards rather than proprietary technologies and thus implementation can be rather inexpensive and widely available. Enterprise applications integration across multiple organizations will be likely to enable inter-organizational business process reengineering and subsequently raise the participating companies to the level of super-efficient company status (2).

In this article, web services are discussed as a tool for integrating various enterprise applications using the Internet as a virtual global network and subsequently for enabling the new design of cross-company business processes and organizational structures. First, web services are explained conceptually with the standard protocols involved. Also, examples are provided as a tutorial for showing how to build and consume web services. Then, business implications of web services will be discussed from the multi-organizational business process perspective.

WEB SERVICES

To implement the inter-organizational business processes, it is imperative to integrate multiple enterprise applications. Enterprise applications in various organizations are most likely to use different software packages, operating systems or machines. Communications among heterogeneous systems have been very difficult. Standards such as CORBA (Common Object Request Broker Architecture) by OMG (Object Management Group) and DCOM (Distributed Component Object Model) by Microsoft have enabled the communications among distributed applications; however, their implementations have been rather difficult and limited.

Since the World Wide Web technology is so prevalent all over the world, it would provide a ready-made infrastructure for communications among the computer systems using its standard protocols such as HTTP (Hyper Text Transfer Protocol). Also, a new standard in representing structured data has emerged. XML (Extensible Markup Language) has become a de facto standard for data interchanges on the Internet. A combination of HTTP and XML provides a basis for inter-application communication protocols and inter-operability among heterogeneous software systems. (See Figure 1).

How to Create Web Service

According to the Word Wide Web Consortium (5), a Web service is a software application identified by a URI (Universal Resource Identifier), whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols. An example of a web service is shown in Figure 2. Web services can be implemented in various ways. Figure 2 shows a source code in C# where each function available is written as a method of a class ‘UnitConversions’. This web service can be accessed at http://ecis.seattleu.edu/courses/ecis569Winter03/LectureNotes/UnitConversions.asmx.

This web service provides functions for converting a temperature in Fahrenheit to Celsius or vice versa. Readers can copy the source code from Figure 2 and save it under <anyFileName>.asmx (e.g., *unitConversions.asmx*) in MS Windows 2000 Server environment. You will see a working web service.

How to Locate and Access Web Service

When you would like to find a fancy French restaurant, you look up a phone directory to locate one to have a nice romantic dinner. If you want to include somebody else’s business logic in your web applications, you need to find one first. The UDDI (Universal Description, Discovery, and Integration) protocol helps locate one like the phone directory for a French restaurant. Refer to a site at http://uddi.microsoft.com/ for an example.

Once you find a French restaurant where you want to have a dinner, you need to know what to order on the menu and how to order one you choose. In a French restaurant, the menu is written in French and you'd better know how to order what you want correctly. To use the web service you located using UDDI, you need to know what to use and how to get the service you want to have. Like French restaurant menu is written in French, the web services are described in XML. Anybody can obtain the detailed description of how to use web services using WSDL (Web Service Description Language). WSDL is written in XML. Figure 3 shows an example of WSDL for the web service shown in Figure 2. A formal definition of WSDL can be found at a W3C site (6).

How to Consume Web Service

Using WSDL, we can get all the information we need to make use of web service methods in our applications without knowing how the web services are implemented on what platforms. This makes distributed applications on the Internet very possible and doable for many organizations. Actual invocation of a web service is transported within the envelope of SOAP (Simple Object Access Protocol) at run time using a proxy (to be discussed later).

Figure 4 shows an example of SOAP for our running example of temperature conversions. Formally, SOAP is defined as follows: SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. (7).

A proxy is like your supplier's branch office located at your site in (let's say) Seattle. Your supplier may be located in (let's say) Taiwan. If you have to contact directly your supplier for parts or other information, it would be very inconvenient. If you can just talk to their representative located in Seattle to do business with the supplier in Taiwan, it would be more convenient and you don't have to worry about long-distance communications. Once you talk to the representative, you expect he/she will somehow contact his/her corporate office in Taiwan. A proxy in web services works just like the representative of your Taiwanese supplier.

When consuming the web service, we need to create a proxy using WSDL. Once the proxy is created, your program can just work with the proxy. All the hard work for communicating with the web service located at some distant location will be taken care of by the proxy.

For example, we can create a proxy source code by doing the following on the Windows environment:

```
> wsdl http://ecis.seattleu.edu/courses/ecis569Winter03/LectureNotes/UnitConversions.asmx
```

This will generate UnitConversions.cs as the proxy source code. Then, you need to compile it like the following:

```
>csc /target:library UnitConversions.cs
```

This will generate a compiled dll (dynamic link library) file (UnitConversions.dll) you can use for your web applications. Figure 5 shows an example program consuming a web service using C# in ASP.Net.

CROSS-ORGANIZATIONAL BUSINESS PROCESS

As Hammer (2) noted, streamlining cross-company processes is the next great frontier for reducing costs, enhancing quality and speeding operations. He continues that it's where this decade's productivity war will be fought. While a tremendous productivity gain was achieved by streamlining intra-company, cross-department processes, a next phase of inter-company business process reengineering will generate even greater productivity gains.

Cross-organizational business processes can be found in supply-chain management, product development, customer relationship management and others. For example, a PC company (e.g., Dell, Gateway, etc.) needs to work with retailers to forecast the demand for different types of PC, hard disk developers to order the components, transportation companies for delivery and so on. How a company collaborates with other companies will determine the scope of success of business operations.

Collaboration of multiple companies requires higher level of integration of business information systems among participating companies. A major obstacle to this is the heterogeneity of computer systems (hardware and software). It cannot be assumed that every company uses homogeneous systems. For example, a company A may be using an ERP (Enterprise Resource Planning) system from SAP on IBM machines while a company B may be using a system from J. D. Edwards. Thus, it is vital to establish an integration environment over the heterogeneous systems. Web services can be a great tool of integration among multiple companies' heterogeneous computer systems.

CONCLUDING REMARKS

Successful integration of inter-company enterprise applications and subsequent cross-company collaboration require strategic leadership and managerial resources as well as technical competency. Many companies that may not have had a great return on investment in their reengineering efforts in the 1990s now have a second chance to make a giant leap toward super-efficiency.

Hammer (2) recommends the following four steps to achieve it:

- Scoping: Identify the appropriate business process to redesign and find partners.
- Organizing: Establishing an executive steering committee and convene a design team
- Redesigning: Design the new, integrated process in a way that fulfills performance goals.
- Implementing: Rollout the new process and communicate.

The web service is one very promising implementation tool to raise the productivity of companies to the next higher level unseen previously. Using the web service, the Internet is truly becoming a business environment where small, agile companies can match larger ones on an equal footing by providing business functions from anywhere in the world.

REFERENCES

1. Davenport, Thomas H. And James E. Short. The New Industrial Engineering: Information Technology and Business Redesign. *Sloan Management Review*, Summer, 1990. 11-27.
2. Hammer, Michael. The Super-efficient Company. *Harvard Business Review*. September – October 2001.
3. Hammer, Michael. Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*. July-August, 1990. 104-112.
4. Kim, Bonn-Oh. Implementation of Business Process Reengineering: Building a Cross-Functional Information Architecture. *Journal of Systems Management*, Vol. 45, No. 12, December, 1994. 30-35.
5. http://www.w3.org/TR/2002/WD-wsa-reqs-20020429
6. http://www.w3.org/TR/wsdl
7. http://www.w3.org/TR/SOAP/

Figure 1. Protocol Stack for Providing Web Services

UDDI (Universal Description, Design, and Integration)
|
WSDL (Web Services Description Language)
|
SOAP (Simple Object Access Protocol)
|
XML (Extensible Markup Language)
|
HTTP (Hyper-Text Transfer Protocol)

Figure 2. Example Source Code of Web Service

```
/* Check the following site for the web service at:
http://ecis.seattleu.edu/courses/ecis569Winter03/LectureNotes/UnitConversions.asmx.
*/

<%@ WebService  Language="c#" Class="UnitConversions"%>
using System.Web.Services;
[WebService(Namespace="ecis")]
public class UnitConversions : System.Web.Services.WebService
{
 [WebMethod]
 /* Fahrenheit to Celsius */
 public double Fahr2Cent(double Fahr)
 {
   return (Fahr - 32) * (5.0/9.0);
 }
 [WebMethod]
 public double Cent2Fahr(double Cent)
 {
   return Cent * (9.0/5.0) + 32;
 }
}
```

Figure 3. Example XML Code of WSDL

```
<?xml version="1.0" encoding="utf-8" ?>
- <definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s0="ecis"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="ecis"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
- <types>
- <s:schema elementFormDefault="qualified" targetNamespace="ecis">
- <s:element name="Fahr2Cent">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="Fahr" type="s:double" />
  </s:sequence>
  </s:complexType>
  </s:element>
```

Figure 4. Example Code of SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /courses/ecis569Winter03/LectureNotes/UnitConversions.asmx HTTP/1.1
Host: ecis.seattleu.edu
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "ecis/Fahr2Cent"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Fahr2Cent xmlns="ecis">
      <Fahr>double</Fahr>
    </Fahr2Cent>
  </soap:Body>
</soap:Envelope>
```

Figure 5. Example Code of Consuming Web Service

* Check http://chinook.seattleu.edu/bkim/temp/webApp1/WebForm1.aspx for this application.

** This program assumes a proxy for the web service has been created.