http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (6): 1127-1131, 2006
ISSN 1812-5638

The Use of Web Services Technology in the Design of Complex Software Interfaces: An Educational Perspective

[1]Ram B. Misra, [2]S. Srinivasan and [2]Dinesh P. Mital
[1]Montclair State University, Normal Avenue, Montclair, NJ, USA 07043
[2]University of Medicine and Dentistry of NJ, 65 Bergen St Newark, NJ, 07107, USA

Abstract: A corporation aiming to implement a sound decision making process must integrate all of its business functions. An essential enabler of this integration is the technology that supports building of various functional (application) interfaces. In this study we show how the Web Services technology can be used to develop functional interfaces of the various modules of a large-scale software project. Two leading web service technologies-Java Web Services Developers Pack (JWSDP) and Microsoft NET were employed to evaluate their ease of installation, programming and usage. It involved the use of the web services architecture in integrating the lesson delivery, project submissions and assessment structure of the course. This study serves as an excellent vehicle for teaching students the principles of functional interface and interoperability not only in the classroom but also as an illustration of new aspects of systems integration methodologies.

Key words: Interfaces, systems, web services, extensible markup language (XML)

INTRODUCTION

Business processes are critical components of all systems that support enterprise-level and business-critical activities. In any large business, there may be hundreds of business processes that need to be managed and integrated with systems in an effective way (Ashby *et al*., 2001). Today, most companies are focusing on ways to increase productivity, reduce cost and seek reliable partnership in order to compete in the intensive e-business market (Aversano *et al*., 2002; Slaski and Coleman, 2003). A corporation's various functional information systems must be able to communicate with each other (Homan *et al*., 2002). The current state of embedded technology in Corporate America consists of many older systems developed on proprietary hardware and software platforms and written in languages that are obsolete. Even though there has been push in corporations-generated from competitive threats-for integration in recent years, the embedded technology has made it quite complex and hence expensive. There is empirical evidence that a big majority of such projects were not successful. In the early 90's, for example, telephone companies tried to replace some of these systems but such efforts were abandoned due to high costs. The proprietary nature of these systems required developing many one-on-one interfaces; this resulted in high costs. The answer is the use of standard software technology independent of hardware platforms, operating systems and programming languages.

Web services technology promises to be one such technology. We decided to perform a study of two competing web services technologies-Java and NET, for the purpose of developing functional interfaces within the context of an online course in software development (Pratt, 2003; Rodgers, 1996). The classroom situation poses some challenges. A programming environment has to be provided to teach and test the programming skills of students who maybe skilled in different languages. Furthermore, there ought to be a single coordinating module of the project (with the instructor perhaps) which will use all of the sub-modules developed by the students in the successful completion of the given project. Lastly any such scenario should be representative of the real-world application of software management (Prady *et al*., 2001; Wood, *et al*., 1998). The onset of XML (Extensible Markup Language) and Web Services (Microsoft's NET and Sun Microsystems' JWSDP) seems to promise an end to the impasse noted above since they both aim and are designed for interoperability amongst different platforms and languages.

MEASURABLE OUTCOMES FOR USE IN STUDENT ASSESSMENT

A successful outcome of the project's applications would be to illustrate the successful learning of the following:

Corresponding Author: S. Srinivasan, University of Medicine and Dentistry of NJ, 65 Bergen St Newark, NJ, 07107, USA

- The ability to install and implement the Web Services Registry which in turn implies the learning of Simple Object Access Protocol (SOAP) messaging (Box *et al*., 2001) and use of API's,
- The ability to implement the database interactivity and the use of the appropriate API in either C++ or Java or any other language,
- The design and implementation of the GUI to query the database through the registry,
- An overview of the complete setup and the way of dividing a large scale software project into several sub-modules and implementing the solution using the Web Services architecture which reflects a real-world application (Christensen *et al*., 2001).

WEB SERVICES-AN INTRODUCTION

Web services are a set of standards to form a Service-oriented Architecture (SOA). This architecture models the interaction between the web services provider, service consumer and the service registries. The initial goal of the web services efforts is to achieve universal interoperability among applications via a set of web standards.

The interoperability standards are developed by organizations like World Wide Web Consortium (W3C). This consortium defines web services as a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format, WSDL. Other systems interact with the web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other web related standards.

The Web Services Interoperability Organization (WS-I), whose 170 members include venders and user companies, announced the availability of its basic profile 1.0 guidelines, which detail how a set of core web services specifications should be used to build interoperable services.

With all the talk about web services, many companies are trying to understand just what these means for their integration strategies. Some believe that the emergence of Web services will displace integration approaches such as Enterprise Application Integration (EAI) or Java 2 Enterprise Edition Connection Architecture (JCA). Using Web services for integration is cheaper and simpler than using EAI and is not limited to Java platform. We believe that web services and JCA will, over time, replace proprietary adapters. But in short term, they will provide a complimentary approach for integration that will co-exist with EAI.

Integration servers such as those from see beyond, vitria and web methods offer solutions that included packaged adapters to major systems, along with toolkits that simplify the process of building custom adapters to other systems such as in-house applications. A small but influential group of Web developers advocate a new approach to Web services. This new architectural approach, called Representational State Transfer (REST), also results in more scalable code. REST developers use URIs to create a common ground so applications can use HTTP and XML to share data. In service oriented architectures, Microsoft has recently introduced the Indigo programming model. Indigo is a collection of guidelines and defines ways of doing tasks meant to simplify the job of security, reliability and messaging in applications that are to be accessed via internet. This version will be released shortly by the Microsoft. It will also make web services applications to communicate with each other regardless of whether the applications are running on windows.

Web services are offered via the web in the form of software performing a function. In a typical web services scenario, a business application sends a request to a service at a given URL using the SOAP protocol over HTTP (Rodgers, 1996). Web services depend on the ability of parties to communicate with each other even if they are using different information systems. XML (Extensible Markup Language) is a key technology in addressing this need. XML tags relate to the meaning of the enclosed text.

WEB SERVICES AND SOFTWARE INTEGRATION

The course entitled Health Care Information Systems Integration, for which this work was done, purports to teach the integration of different modules making up a health care related information system spread across an intranet or the internet. Our software project consisted of four modules (Fig. 1), each of them implemented in whichever platform and whatever language the student is skilled in. Each of these modules will be 'wrapped' in an SOAP object and these objects are transmitted from one station or computer onto another in the form of XML. At the recipient computer, that of the Instructor, the sub-modules in the guise of SOAP messages are suitably transformed and used as if they were written on the same computer in the same programming language and within the same operating system (Pratt, 2003).

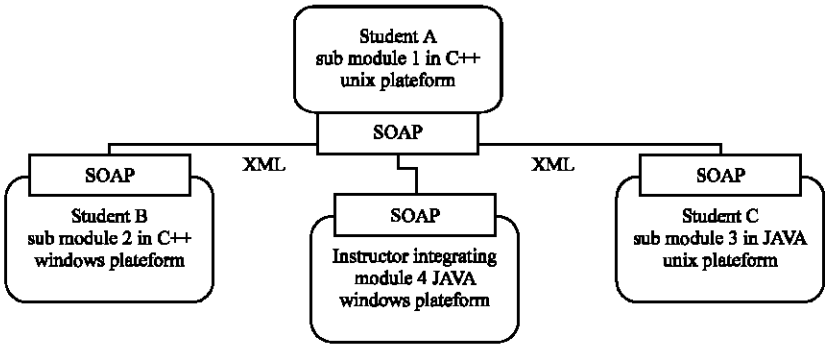

Fig. 1: A schematic outline of interoperability using web services

WEB SEDRVICES USING THE NET TECHNOLOGY

Microsoft's NET platform seeks to blur the boundaries between platforms, systems and modular applications. It offers the ability to develop applications in multiple languages. It gives the developer the potential to program in his or her most proficient language. It also enables systems to take advantage of each language's distinct competencies. While using NET to develop code for this study, we observed first hand, the role of NET in system interoperability and exactly how easy NET makes the task of developing applications capable of cross-platform communication and data processing. To start with one must first determine, based on the needs of the study, the Web Services the study intends to consume. Next, one must inform NET using the procedure supported in NET. The Web Service Description Language, WSDL, files describe the input and output parameters of the Web Service (Christensen *et al*., 2001; Bellwood *et al*., 2002). They are simply XML files that explain how the interaction between the consumer (our study in this case) and the Web Service will occur. If a Web Service accepts an integer and string and returns a Boolean, the WSDL informs NET of this. Since these files are XML files, it does not matter what platform they reside on. WSDL files can be on Windows, Linux, UNIX or any other kind of platform.

Implementing an external Web Service into an application is as easy as adding a reference in the project. These references to Web Services are called, "Web References" and must point to the WSDL files on a remote or local server. The programmer supplies NET with the URL of the WSDL file and NET is returned a summary of the methods that this Web Service offers. Now, the functions of that web service may be called as simply as any other function of the local code.

WEB SERVICES WITH JWSDP

Sun microsystems introduced Java as a fully object oriented and portable platform-independent programming language which made it an ideal choice for developing the web services environment around it. This choice is even more attractive as the new Java APIs for XML (JAXP, JAXM, JAXRPC) are now fully available. The Java APIs for XML define strict compatibility requirements to ensure that all implementations deliver the standard functionality. JAXP code can use various tools for processing an XML document and JAXM code can use various messaging protocols on top of SOAP. Besides data portability and code portability, the Web services infrastructure includes features such as security, distributed transaction management and connection pool management, all of which are essential for industrial strength web services (Basiura *et al*., 2002; Dournaee and Dournee, 2002). Because XML and the Java platform work so well together, they were also are very apt to procure information from a database for queries originating from different 'clients' spread across the network. For assessment, we used the ideas discussed by Pratt (2003). As part of their assessment the students were asked to develop a software project which implemented a 'client-server' networking task. Figure 2 shows how the mechanisms of XML, SOAP and Java Web Services Pack were used in this study.

The computer acting as a database server contains an Oracle Database with biomedical data. It has registered its 'services' with the Web Registry (part of the Java Web Services Pack-JAXR) contained in a Windows NT server. The other computer acts as a client who wishes to seek the information contained in the database and 'looks up' in the Web Registry. Upon a successful search the client and the database server will interact with one another, though the whole thing was affected automatically by means of the Web Registry and the Web Services Pack.

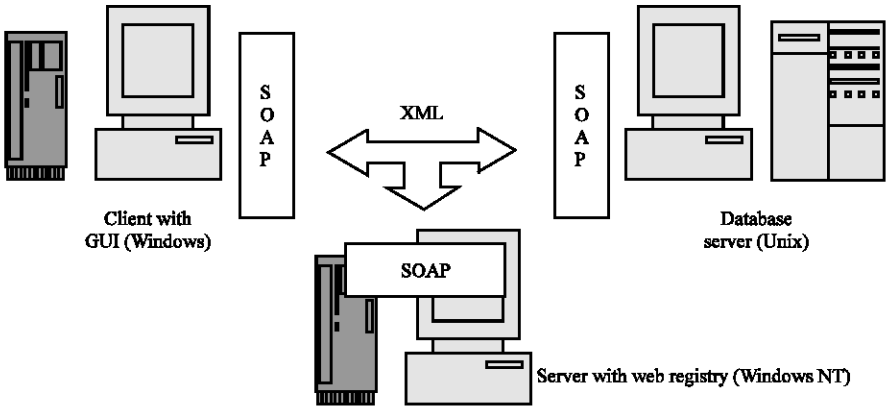

Fig. 2: Illustration of a client-server network using web services

RESULTS

Upon obtaining feedback from students as to the efficacy and ease of use of the two competing software, we learnt that NET offered an unmatched ease-of-user. As has already been described, when dealing with Web Services, many things need to be considered and NET takes care of this tedious work. For instance, XML, SOAP and HTTP need to be taken into consideration because their interactions are essential to Web Service use. However, NET simplifies their interactions for the developer. The JAVA alternatives to NET-J2ME, J2SE and J2EE-all require the necessary SOAP work be done by the developer whereas NET does much of this work for the programmer. When discussing NET's role with Web Services interactions, it is comparable to the WYSIWYG editors' role with web site development. In NET, the XML that makes up the SOAP message never has to be seen or edited by the developer. It all gets handled behind the scenes, keeping the code clean, efficient and easy to navigate. If the developer requires greater control of the SOAP message, he/she may also access it for editing.

Therefore, the answer as to which software development system is better very much depends on the a priori knowledge of the mechanisms involved in the web-services programming. Students who were already aware of such details were found to greatly benefit from the very user-friendly and easy to install Visual StudioNET while those who were in the process of learning the theory benefited better in using JWSDP. In either case the students on the whole learned a lot more complex information within a short time than they would have otherwise in a traditional software programming class structure. This novel strategy for teaching software development using the web-services architecture served not only to make the students aware of the state-of-art programming paradigms and developments taking place in the software engineering arena but also simultaneously motivate them in learning the relevant theoretical concepts in a more diligent manner with a greater comprehension of their role in the real world software systems.

CONCLUSIONS

In this study we illustrate how the Web Services technology can be used in developing a dynamic interchange in the setup and execution of the various modules of a program. Also the study not only purports to teach interfaces programming but also to illustrate its relevance in a real-world application. Thus the students would be learning all those aspects of the JAVA or C++ programming language relevant to solving a real world problem and at the same time use the latest aspects of the language such as the XML and SOAP interactivity and Web Services design and development. Furthermore, we explored the efficacy and ease-of-use of two competing web service development technologies, namely, Microsoft's NET and Sun Microsystems' JWSDP. The results of this pilot study are very promising and work is currently underway to implement the web services paradigm for more courses. The pedagogical experiment described in this report serves as a valid and an important observational study in the use of web-services in education and in practice.

REFERENCES

Ashby, J.V. *et al*., 2001. A multidisciplinary scientific data portal. High Performance Computing and Networking, 2110: 13-22.

Aversano, L. *et al*., 2002. Integrating document and workflow management tools using XML and web technologies: A case study. Proc. 6th Eur. Conf. on Software Maintenance, Reeng., pp: 24-33.

Basiura, R. *et al*., 2002. ASPNET Security, WROX Publication, pp: 1-55.

Bellwood, T. *et al*., 2002. Universal Description, Discovery and Integration specification (UDDI) 3.0. http://uddi.org/pubs/uddi-v3.00-published-20020719.htmj.

Box, D. *et al*., 2001. Simple object access protocol (SOAP) 1.1, 2001. http://www.w3.org/TR/SOAP/.

Christensen, E. *et al*., 2001. Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl/.

Dournaee, B. and B. Dournee, 2002. XML Security. McGraw-Hill, ISBN 0072193999, pp: 12-20.

Homan, D., Kalavagunta, S. and C. Klima, 2002. Web Services and Integration: Use integration approaches that provide the right balance of simplicity, flexibility, scalability and reliability. Information Week, October Issue.

Prady, S.L., D. Norris, J.E. Lester and D.B. Hoch, 2001. Expanding the guidelines for electronic communication with patients: A pplication to a specific tool. J. Am. Med. Inform. Assoc., 8: 344-348.

Pratt, D.S., 2003. Introducing Microsoft NET, 3rd Edn., Microsoft Press.

Rodgers, R.P., 1996. Java and its future in biomedical computing. J. Am. Med. Inform. Assoc., 3: 303-307.

Slaski, B. and G. Coleman, 2003. Accelerated Information Sharing for Law Enforcement (AISLE) using web services, Int'l Conf. Web Services, Las Vegas, USA., pp: 1312-1318.

Wood, F.B., V.H. Cid and E.R. Siegel, 1998. Evaluating internet End-to-end performance: Overview of test methodology and results. J. Am. Med. Inform. Assoc., 5: 528-545.