Enhancing E-Learning through Web Service and Intelligent Agents*

Nasir Hussain and M. Khalid Khan

College of Computing and Information Sciences, PAFKIET, Karachi, Pakistan

nasirwt@yahoo.com, khalid.khan@pafkiet.edu.pk

Abstract

E-learning is basically the integration of various technologies. E-Learning technology is now maturing and we can find a multiplicity of standards. New technologies such as agents and web services are promising better results. In this paper we have proposed an e-learning architecture that is dependent on intelligent agent systems and web services. These communication technologies will make the architecture more robust, scalable and efficient.

I. Introduction

In this paper, we have presented an e-learning architecture based on web services and intelligent agents using a service-oriented approach. The advent of the internet gives us new challenging possibilities for teaching and learning. The World Wide Web makes it possible to develop educational content in a new perspective. The terms e-learning, web-based learning and hypermedia aided learning have become virtually interchangeable for online education.

E-learning systems use various tools and technologies. Several sets of various standards and specifications have been developed in this context. ADL (Advance Distributed Learning) SCORM (Sharable Content Object Reference Model [SCORM (2004)] is one of the better known models for learning content objects. It provides interoperability between different learning systems by means of standardized packaging, sequencing and communication methods for learning contents.

---

* An earlier version of this paper was presented at a conference on Information and Communication Technologies organized by Institute of Business Administration in September 2005.

To achieve interoperability between different systems across the internet, web service technology now plays an important role. It is mainly used for distributed web applications that uses communication protocol based on 'Simple Object Access Protocol' (SOAP) [Robert (2003)]. It frees the client and server from any platform-specific implementation restrictions, allowing much greater flexibility in a heterogeneous computing environment. We have designed an architecture using web service based intelligent agents enabling users to consume educational services of each other in e-learning networks and thus enhance the learning experience of individual learners, teachers and authors through user agent interactions.

Intelligent agents have been suggested as a technology to deal with challenges such as the increased need for personalization, high latency, demand for large transfers and disconnected operations. They are mainly used in distributed environments [Wang (2003)].
The paper is organized as follows. Section II discusses developments in E-Learning. Section III and IV are about web services and intelligent agents respectively. Section V discusses our proposed architecture. In section VI we discuss benefits of the proposed architecture and section VII concludes.

II. E-Learning

Use of technology to enhance learning not only involves technologists but authors, learners, teachers and instructional designers. ADL plays an important role in e-learning and in building SCORM to deploy learning contents on any learning management system that conforms to SCORM standards. ADL integrates the work of various standards and specifications such as IEEE (Institute of Electrical & Electronics Engineering) LTSC's (Learning Technology Standard Committee) LOM (Learning Object Metadata), AICC (Aviation Industry 'Computer Based Training' Committee) CMI (Computer Managed Instructions) data model, IMS (Instructional Management System) Content Packaging and many others [SCORM (2004)]. SCORM provides the foundation for using standardized based learning contents that can be deployed to any LMS and student's progress can be tracked by the standardized method using the AICC CMI data model. ADL is now expanding its e-learning research and currently work is proceeding on specifications for object repositories, gaming, performance aiding and intelligent tutoring systems (http://www.adlnet.org).

Online learning is learning via the web. In this type of learning a learner is connected to the internet and utilizes educational contents through web browsers. As a person can take some information about a company from a company's web site so can a learner obtain a class lecture from e-learning services. Learning Management Systems (LMS) can provide learning contents, search of contents and course management methodologies. There are several vendors of LMS in the market, which support the SCORM standard using its Run-Time Environment (SCORM RTE) specification. We can also develop an architecture for SCORM compliant LMS using SOAP and .NET XML web services [Nigel (2003)]. Another usage of .NET XML web services is in the form of MIT's 'iCampus' (http://icampus.mit.edu) and 'iLab' (http://icampus.mit.edu/ilab).

The major advantages of online learning are that it is accessible anytime and anywhere. A learner does not only see course material in the form of a web page. He is introduced to a learning network, which provides much greater learning facilities. E-learning can provide facilities such as personalization, learning forums, digital libraries, virtual labs, authoring services, virtual classes and many others. E-learning technology needs support from distributed internet technologies that can drive it over the world wide web. In the following section we will discuss how distributed technologies like web services and intelligent agents can play a functional role in improving e-learning infrastructure.

III. Web Service

Web service is a technology that has been developed to provide various types of services over a web. The main advantage of using a web service technology is cross-platform communication. At present (i.e in 2005) there are two major competitors in web service technology Microsoft and Sun. As far as implementation is concerned both use common standards and protocols, such as Simple Object Access Protocol (SOAP), Extensible Markup Language (XML), Web Service Description Language (WSDL) and Universal Discovery Description & Integration (UDDI) [Roger (2001)].

SOAP is an XML-based message exchange protocol that is used to communicate between web services and their clients [Aaron (2003)]. With the help of this lightweight protocol we can easily exchange structured information in a decentralized distributed environment.

WSDL provides description of a web service. Each web service has a WSDL file which is basically an XML file that describes a set of SOAP messages and how the messages are exchanged between web services and clients [Roger (2001)].

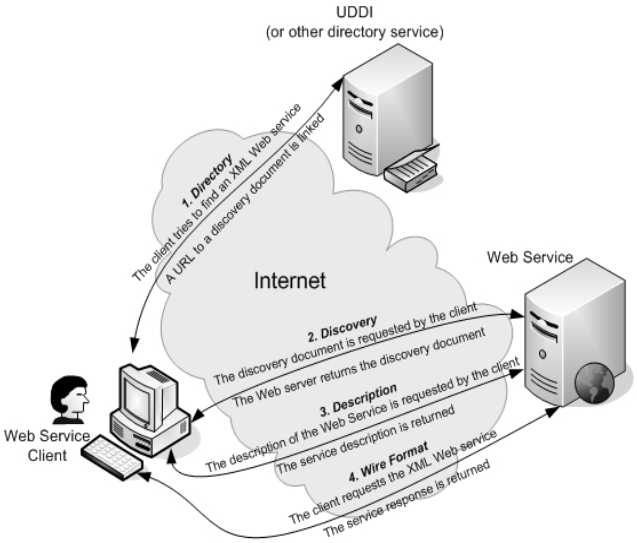
UDDI is often called the Yellow Pages of web services [Roger (2001)]. A UDDI is a directory of web services having XML files describing a business and the services it offers. We will use UDDI in our architecture.

III.I. Web Service Infrastructure

In a web service-based computing model, both clients and the web service providers are unaware of implementation details. If the client wants to consume a web service, he will have to go through four stages. These four stages are directory, discovery, description and data which is also called wire format [Roger (2001)].

Figure 1 presents a web service infrastructure. At the first stage (directory), a client is searching for a web service. Directories services such as UDDI provide a central place for storing published information about web services. The client searches a directory and finds a URL. In the second stage a client sends a request for service description documents. The server returns the discovery document that enables the client to know about the presence of a web service and its location. In stage three the client sends his request for a particular web service. The service description is sent by the server in XML format which specifies the format of the messages that the web service can understand.

Figure 1
Web service infrastructure



Finally at the last stage the client requests the XML web service and is enabled to utilize it. The server sends the required response to a client. To enable communication between disparate systems web services use open wire formats. Open wire formats are the protocols that can be understood by any system that is capable of supporting common web standards, such as HTTP and SOAP.

VI. Intelligent Agent

An agent, intelligent agent or agent-based system is a software-based computer system that has several properties such as; autonomy, social ability, reactivity, goal-directed, mobility and collaboration. Agents are used to make a user's tasks easier in a distributed network environment [Anselm (1995)]. Every agent can perform tasks it is programmed for. Agents can communicate with users as well as other agents in a network. Agents are also helpful in exchanging data in cross-platform systems.

Mobile or intelligent agents are also software-based agents that have the ability to move in a network, seeking information for their owner and negotiating and concluding business deals [Anselm (1995)].

These intelligent agents can be useful for many applications. Mostly intelligent agents are used to gather information for users according to his/her demands and matching user profiles. Another advantage of using intelligent agents for mobile devices is that they can detect the capabilities of mobile devices such as memory, storage capacity and processing power so that they can work accordingly [Wang (2003)].

Applications of intelligent agents are not limited to general information retrieval from network. They can also be used for network management [Khalid (2005)], electronic commerce, application maintenance and many others purposes. Interface agents are one of the types of intelligent agents that are commonly designed today. They have basically a graphical user interface character that has an ability to learn about user preferences, interests and habits. They can interact with users audio-visually to maintain user attention with respect to a specific task [Amit (2001)].

We used intelligent agents in our e-learning architecture for online education. Users may be learners, teachers or authors. For a learner, intelligent agent can find information about his learning needs such as books, courses, virtual classes, labs and learning forums etc. in a learning network. For teachers' agents can provide information about teaching materials, learning contents, books and forums using user profiles. In our architecture, the applications of intelligent agents not limited to these areas. Agents may also play an important role in integrating other e-learning network capabilities and develop scalable and interoperable architectures. To substantiate our view that the use of an agent in the system will be efficient, we are outlining here some agent properties.

Autonomy: This is the ability to act and make decisions without being controlled by anyone else. An autonomous agent is a system situated within and a part of an environment that senses the environment and acts on it over time in pursuit of its own agenda so as to affect what it senses.

Communication: One of the key properties of agents is the ability to speak with humans (interface agents), and/or devices. Communication systems between agents (languages) frequently used are:

- KQML: Knowledge Query and Manipulation Language. This is a language and protocol for exchanging information and knowledge using “performatives”.
- KIF: Knowledge Interchange Format. This is a computer-oriented language for the interchange of knowledge among disparate programs.

Collaboration/Cooperation: Agents are collaborative when they are able to work together. The agent is able to communicate and negotiate with others. This language is deliberative and may coordinate its actions with others. Collaborative agents are useful among other things when a task involves several systems on the network. Negotiation is the main concern of the collaborative agents. When coordination occurs without collaboration, it needs negotiation.

Deliberation: This is the ability of understanding rules and applying them without waiting for instructions. “This involves internalizing an explicitly represented symbolic model of the world, in which decisions... are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulation” [Wooldridge (1995)].

Mobility: Since Java appeared, we find a lot of mobile agents which can move in a network seeking out some information or completing tasks. Mobile agents can move from one system to another system in the same network. They also have the ability to move in another network. One of the main concerns with regard to mobility is the potential security weakness of mobile agents, since they because it can also be used to run malicious codes in a network.

Learning: Agents can learn not only from user inputs but also from other agents in a network. They can modify their behavior by utilizing new knowledge. Despite the fact that learning is an important factor in intelligence acquisition, there are only relatively few agents able to learn; most of the time they have fixed (pre-compiled) rules and knowledge bases.

Pro-activeness: Pro-active actions are intended to cause changes, rather than just reacting to change'. Pro-active agents generally follow plans, or at least execute rules when the environment reaches a known threshold. Sometimes “pro-activeness” is used with the same meaning as “deliberation”, but an agent may be pro-active, when it is requested to perform pro-active tasks, as opposed to deliberative agents, which themselves decide to be pro-active.

Planning: The agent organizes by priorities the actions to perform during its life. For many researchers planning is one of the most important properties an intelligent agent possesses. Planning is used by deliberative and pro-active agents according to their knowledge of the environment and the possible actions to apply to it.

Delegation: An agent may ask someone else to perform one of its tasks. This capability is very important since it facilitates resource balancing.

V. Proposed Architecture

Our e-learning architecture has two components, client-end and server-end. Both ends are connected through the internet with common sets of internet protocols such as HTTP, SOAP and XML

Various types of computer devices are present in the client-end system which uses diverse platforms. On the other hand the server-side system uses a single platform that must communicate in a heterogeneous environment using internet standard-based protocols and

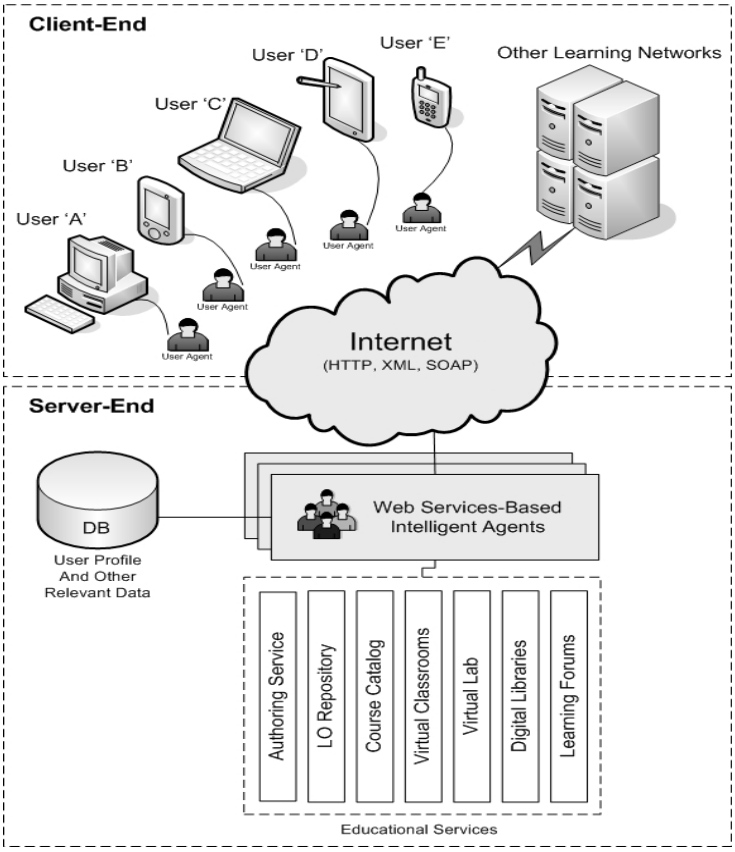
provide services in a scalable and robust manner. We have suggested a 'user agent' on the client-side for customization and assistance in learning whereas the server-end consists of web service based intelligent agents. These can also be mobile agents for search and discovery in other e-learning networks. Below we have broadly described our client-end and server-end systems.

V.I. Client-End

In Figure 2, it is clear that the client-side system contains diverse computing devices and platforms. Some users may use desktop PCs, others may use handheld and wireless devices to access learning services. We have suggested an interface agent that works on the client-side.

This interface agent can be run on any platform and its versions must be different according to computing device hardware profiles such as its memory, processing power and storage. In this case, we have suggested that the JAVA platform is appropriate for developing interface agents so that they can be run on any computing device whatever its platform may be. We have categorized computing devices to make different versions of user agents for each category. The category of computing devices may be a desktop PC, PDA, tablet PC and mobile phone etc.

Figure 2
High-level view of e-learning architecture



The user agent would be configurable by the user who would specify frequency of user agent and server agent contact, processor usage, memory usage and network bandwidth etc. Thus the user agent can give an optimized performance according to these configurations. The user agent has also routines to handle communication between web service based agents on the server side, as well as gathering other relevant learning materials from the web. SOAP is used for communication between the user agent and the web service based agent. A user agent would query information from the server-side agent on the basis of the user profile and the device profile. User profile specifies user age, area of study, interest etc. whereas device profile specifies processing power of computing device, memory, storage etc. The user agent also has animation and speech characteristics to interact with the user and maintain user attention during learning. The user agent would also be helpful for teachers and authors when they are developing leaning contents. It can suggest keywords for LOM and other learning materials such as related SCORM assets by searching asset metadata on servers so that the reusability of learning content may increase.

V.2. Server-End

The server-end system of our e-learning architecture is composed of several components that includes web service based intelligent agents, database and educational services. The main focus of our architecture is on web service based intelligent agents which can make our architecture robust, scalable and interoperable. We incorporate these agents with web services and elaborates the main benefits of using web service based intelligent agents. These agents can use SOAP, XML and HTTP to communicate over a network. They can depict their functionality using WSDL as these are based on web service. Clients are enabled to call our agents using standard based SOAP messages.

Learning Management Systems in other networks can be connected with our web service layer and access our learning object repository. Anyone can search for learning objects and assets using its metadata through SOAP-based messages. The main goal of ADL is reusability of learning contents in a distributed network environment. We have increased the reusability of learning contents as well as other learning services with the help of these web services.

The database is used to store user profiles and other relevant data and it can be accessed through web service. Our agents can use the database to store relevant information about authorized users using their user profiles when they are not online. From this point of view our intelligent agents can work in a disconnected environment and serve stored information to users while they are online. If a user asks for some educational service then our intelligent agent can be mobile and move in a network to discover the requested educational service for the user. Intelligent agents can track students due dates of assignments. They can also search for relevant news items, learning objects, books, forums, courses, classes, labs etc. and track newly added services and advertise them to respective clients and web sites. Not only individual clients may avail these services but other e-learning networks will be enabled to access our intelligent agents and educational services using SOAP.

VI. Benefits

By integrating web service and intelligent agent technologies we can achieve many benefits that cannot be accomplished from common methods. Some of the major benefits we found in our architecture are as follows.

***Service-Oriented*:** The title of the paper suggests that our e-learning architecture is based on a service-oriented approach. Using web service and intelligent agent technology we can provide

educational contents in the form of different services such as learning object repository, digital library, LMS, virtual classrooms, virtual labs, authoring services etc. The user can add as many services as he likes by adding his web service layer.

***Interoperability*:** One can develop web services using JAVA or .NET platforms; both communicating using standard sets of protocols such as HTTP, XML and SOAP. One can easily integrate his e-learning system with other systems in this diverse web-based environment.

***Reusability*:** The reusability of learning contents would increase with the help of web services. Any learning management systems can access our learning object repositories and other educational services, and utilize it in his own way through SOAP-based messages.

***Scalability*:** Our architecture provides a scalable environment where the user can develop his own services easily without interrupting the core architecture. The user can add new educational web services and register his service using UDDI or other directory services.

***Maintainability*:** Maintainability of educational services is facilitated because educational services are divided with respect to category and functions.

***Flexibility*:** Agents are always flexible as they can move in a network to find information; our web service based agents can communicate with other web services in a network using standard-based protocols such as SOAP.

***Customization*:** A client can customize his 'user agent' by configuring memory usage, processor usage, frequency of getting information from server agent and many other things. User profile also plays a part in customizing user interests.

***Interactivity*:** The main purpose of a user agent on the client-side is to provide interactivity with the user. A user agent would be helpful in maintaining user attention towards learning objectives as well as developing learning contents.

There are several other benefits that can be realized after implementing this e-learning architecture and it will enhance e-learning experience.

VII. Conclusion

We have proposed a service-oriented architecture for e-learning that would increase interoperability, reusability, scalability, maintainability, flexibility, customization and interactivity in a heterogeneous e-learning environment. Using web services and intelligent agent technology it is easy to program and maintain services at the server-end. User agents on the client-side can perform several tasks. This contributes to the achievement of the ADL aim of

reusability for SCORM between distributed LMS. Learning objects as well as assets can be easily accessed by authors and instructional designers.

We are working on the implementation of our proposed architecture to realize its full benefits and expanding our architecture by using agent technology with SCORM complaint learning objects.

REFERENCES

Aaron S. (2003), Understanding SOAP, MSDN Library Articles, Microsoft Corporation. http://msdn.microsoft.com/webservices/default.asp

Amit (2001), Application Maintenance Using Software Agents, Proceedings of the First International Workshop on Source Code Analysis and Manipulation, IEEE.

Anselm L. etal (1995), An Infrastructure for Mobile Agents: Requirements and Architecture, http://citeseer.ist.psu.edu/lingnau95infrastructure.html.

Khalid etal (2005), Multi-Agent Architecture for Distributed and Dynamic Network Management, Proceedings of the 1st International Conference Frontier of Information Technology.

Nigel L.T etal (2003), Using SOAP and .NET Web Services to Build SCORM RTE and LMS, Proceedings of the 17th International Conference on Advance Information Networking and Applications, IEEE.

Robert I.C. (2003), Designing Enterprise Applications with Microsoft Visual Basic .NET, Microsoft Press.

Roger W. (2001), XML Web services Basics, MSDN Library Articles, Microsoft Corporation. http://msdn.microsoft.com/webservices/understanding/webservicebasics/default.asp

SCORM (2004), Advance Distributed Learning, Sharable Content Object Reference Model Overview 2nd Edition, http://www.adlnet.org.

Wang A.I. etal (2003) " Mobile Agent Architecture for Heterogeneous Devices", IASTED International Conference on Wireless and Optical Networks.

Wooldridge M. etal (1995), Intelligent agents: Theory and Practices, , Knowledge Engineering Review. http://citeseer.ist.psu.edu/context/1465/0