Development and Integration of Expert Systems Based on Service-Oriented Architecture

CHUNG C. CHANG CHIEH-TAO TSENG
Department of Information Management
Chinese Culture University
55 Hwa Kang Road, Yang Ming Shan, Taipei
TAIWAN
http://www.pccu.edu.tw

Abstract: - Service-Oriented Architecture (SOA) is one of the technologies under extensive discussion in recent years, and most noteworthy of all are Web Services. Web Services utilize core technologies, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI), to construct an easy cross-heterogeneity platform, an information system that can be easily reused and expanded.

Based on the SOA theory, the initial goal of this study is to develop a computer network problem diagnosis expert system for resolving frequently seen Web issues. Utilizing ASP.Net and SQL Server as the software tools, and based on Web Services, it provides for users a self-diagnostic computer network problem expert system and then utilizes this system to incorporate knowledge bases, expert systems, and conventional information systems developed in different development environments in order to expand the original system function and verify one of the most significant advantages of Web Services – communication between heterogeneous systems.

Key-Words: - Service-oriented architecture, SOA, Web services, Expert systems, XML, Network problem diagnosis, Computer networks, System development and integration

1 Introduction

Since the development of artificial intelligence in the 1950s, the expert system has been one of the extensively-discussed issues of artificial intelligence. Besides the more famous MYCIN (medicine), XCON (business), and CATS (industry) during the early days, the expert system is also recurrently applied to agriculture, biology, weather forecast and teaching. In response to the surfacing of computer networks, the expert system has also gradually moved from the stand alone system in the beginning to the Web-based system. Yet knowledge acquisition difficulties and logical deduction accuracy concerns have prevented development of the expert system from more significant breakthroughs. This study will lead the expert system into a popular realm – Service-Oriented Architecture (SOA). Based on the concept of distributed computing, Web Services program design is known for its loose coupling and open standard. It plays a very significant role in promotion of the expert system, simplification of system integration and enhancement of system functions, reduction of system development cost, and knowledge sharing. The following is a brief introduction to the areas covered by this study.

2 Services-Oriented Architecture

SOA is a popular system architecture model. By its name, one can tell it is a framework primarily for providing services. If we say Web Services are the cement, then SOA will be the reinforcement of the building. Jian [5] in his article "Applications of Service-Oriented Architecture" mentions that: According to its main concept, SOA is a set of software elements put together in response to corporate requirement. The constituents normally include three portions: software elements, services, and process. When a corporation is faced with external requirements, the process defines the steps for handling external requirements, services include all program elements of specific steps, and the software elements are responsible for executing working programs. He believes SOA has the following technical characteristics:

(1) Distributed framework

SOA consists of systems from different parts of the network. They can come from local area networks or wide area networks. For example, Web Services is SOA that utilizes Internet HTTP Protocol for linkage. This approach allows Web Services technology to quickly become one that can be used by all system

platforms that support the Internet.

(2) Loosely coupled interface

Traditional systems mainly sever functional requirements of application systems into interrelated parts, modules, objects or elements. It takes great efforts of the developer to understand how the parts are designed and used to ensure that the linkage condition of the components is not violated. In such a case, replacing the original design with different components becomes a difficult task. The approach of SOA is that it organizes systems via interface standards. As long as the interface requirement is met, the parts can be replaced as one wish. This significantly enhances system change flexibility.

(3) Based on open standard

The use of open standard is the core feature of SOA. The fact that past software element platforms such as CORBA, DCOM, RMI and J2EE, employ exclusive protocols as element linking standard make communication between elements of different platforms impossible. SOA's emphasis on open standard and interaction avoids integration troubles among programs developed by different platforms (e.g., .Net Web Services and Java Web Services).

(4) Process centric

System construction begins with understanding process requirements of specific tasks. The process is severed into service interfaces (including data input and output formats). This way, other developers can develop or select appropriate elements according to the service interface to complete the task.

At present, leading SOA vendors include Microsoft, BEA, IBM and Sun. Primary applications are presented via Web Services and developed through JAVA or ASP.Net. BEA [7] in its technical whitepaper, IT Transformation to Service-Oriented Architecture, mentions that the main strengths of SOA can be divided into three categories: low-cost integration, enhancing system transparency and independence, and industry-compatible service level.

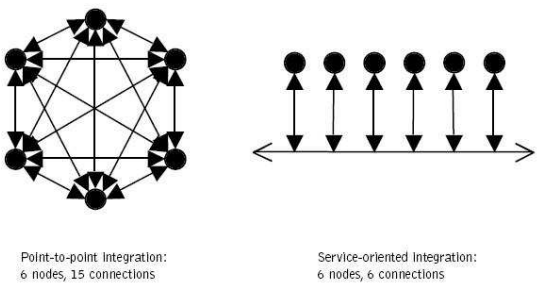

Figure 1 Service-Oriented Integration Reduces Complexity [7]

Baglietto et al. [6] identify three basic reasons for choosing SOA:

(1) Universality: Enterprise information systems utilize integration and coordination of HTTP or SMTP.

(2) Cross-platform: Cross-platform feature that uses XML for document exchange and access services via SOAP.

(3) Distribution requirement: Distributed central registration; UDDI standard employed for the architecture.

3 Web Services

The primary framework of Web Services comprises four core technologies: XML, SOAP, WSDL and UDDI [4]. Our main concern is XML message transmission. XML is a standardized document. Web Services owes most of its communication and integration to XML, which provides description, storage and transmission formats for Web Services for data exchange.

According to the definition of W3C [3], Web Services is an application program that utilizes URI (Uniform Resource Identifiers) to identify documents in the web environment and allows download of files, services, and e-mail boxes. The interface and linkage of this application program employs XML for definition and description. According to this definition, Web Services can be discovered and used by other agent programs and triggered by XML messages sent via Internet protocols. The job of Web Services is not to create new application programs, but to release services provided by completed application programs, so other application programs will have access to the same services.

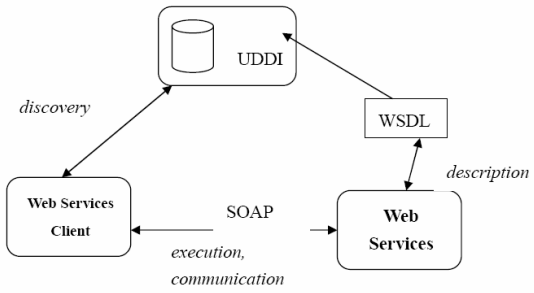

Figure 2 Web Services Technical Framework [3]

4 Expert Systems

Expert systems refer to the fact that expert's professional knowledge and problem solving approach are extracted and turned into an information system by

knowledge engineers to help users solve problems [2]. As the stand alone system of the early days was transformed into networks in recent years, expert systems were also networked [9]. It gave rise to development of the shells of many knowledge base tools and expert systems [11] and makes expert system development easier.

Development of expert systems can be divided into five stages: conceiving stage, birth stage, establishing stage, growing stage and maturing stage.

The time before 1960 was the impregnating stage of the expert system. The research focus of this period of time was General Problem Solver (GPS), yet the attainment was limited. In 1958, J. McCarthy proposed the AI language LISP [10].

The time period of 1960~1970 was the birth stage of expert systems. In 1965, Stanford University developed an expert system Dendral, which is regarded as the first expert system, for analyzing molecular structure in chemistry. In 1965, L. A. Zadeh proposed the fuzzy logic to give expert systems more theoretic foundation.

The year of 1970~1980 was the system establishing stage. In 1972, Edward Shortliffe at Stanford University developed MYCIN, an expert system for diagnosing blood infections. In 1975, M. L. Minsky presented the frame-based knowledge representation method. In 1978, Carnegie Mellon University and DEC jointly developed R1, which also termed XCON, sent expert systems into the realm of business application.

The years of 1980~1990 marked the growing stage in which the expert system made its debut in business application. In 1983, InteliCorp unveiled KEE, an expert system construction tool that incorporates diverse knowledge representation and inference methods. Since 1990, expert systems have been gradually applied to different disciplines, such as agriculture, medicine and education, and further integrated with the Internet. Further, through incorporation of expert knowledge in different fields, expert systems have been frequently employed by various forecasting and analytical investigation systems [8] [10].

The primary framework of the expert system can be divided into six parts:

(1) Knowledge base: The core component of expert system. It stores the knowledge that experts solve problems with.

(2) Inference engine: The mechanism used to handle and control the inference process of expert system. According to the problem and knowledge base content, logic deduction is employed to find solutions.

(3) Working memory: Temporary data storage area used to store problem-related data and intermediate-results generated during the inference process.

(4) User interface: Used to provide for users friendly, graphic-based communication interface to enable them to communicate with the expert system.

(5) Knowledge acquisition interface: Used to convert expert knowledge and experience extracted by knowledge engineers into frameworks that can be used by the computer. It can also be used for modifying and updating the contents of knowledge base.

(6) Explanation subsystem: Used to provide explanations for user's questions. The most common explanation functions are "Why" and "How".

5 Development of a SOA-Based Expert System

In order to further demonstrate the integration of expert systems and other heterogeneous systems, we first develop a SOA-based expert system prototype. The following paragraphs explain major system frameworks and related functional modules.

(1) Integration of the Expert System and Web Services

The expert system established for this study plays the role of a Web Services provider, which furnishes services to Web Services application while Web Services application provides expert system services for the Web Services requester of the Client end. On the other hand, the Client end submits service requests to the expert system via Web Services application.

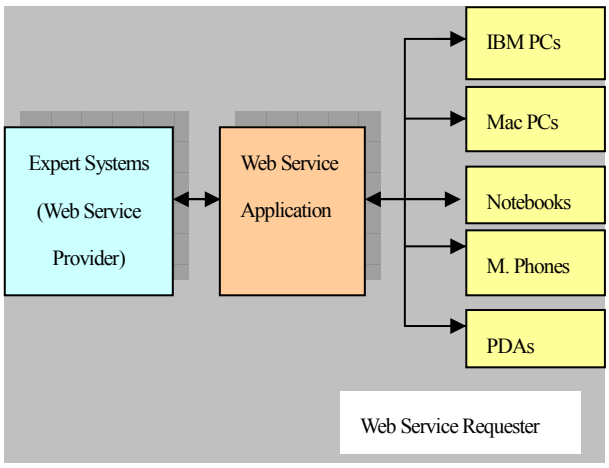

Figure 3 The Expert System Plays the Role of a Web Services Provider

(2) Method to Generate the Recommended Solution

The knowledge base of the expert system is structured and stored via SQL Server. The inference engine retrieves knowledge or rules from the knowledge base via SQL Server and records user's choice, which is then sent back to the inference engine to generate a set of recommended solutions.

(3) Knowledge and Rule Modification

The knowledge base of the system is established via SQL Server, which makes modification very easy. Modification of knowledge content requires only login to SQL.

(4) Inference Engine

The inference engine of the system employs the forward chaining method. Through Q&A it collects various data related to the problems that the user currently encounters and infers the conclusion accordingly. For instance, user logs in via ADSL, gives ping command to DNS or gateway and receives no response. The ADSL link light of the modem is not on, and the upload or download light indicates no response either. Through the information above, it can be inferred that the modem may have broken down or the ADSL mechanical room may have experienced troubles. It is recommended that the unit be delivered for repair or ADSL operators be called for an inspection.

(5) User Interface

Front-end user interface employs the Q&A method. Each question is a multiple-choice question with two or more choices. After a series of Q&A, the recommendations and answers generated by the inference engine are presented to the user on the screen for reference.

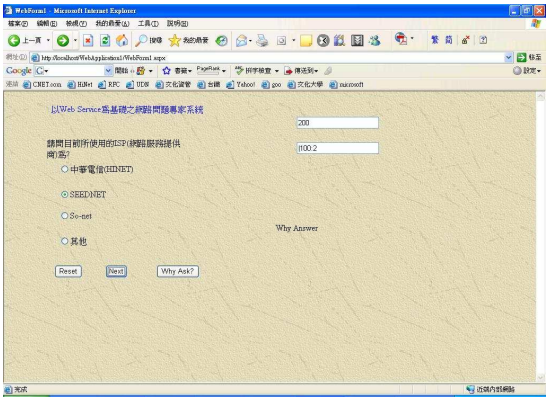

Figure 4 User Interface

(6) Explanation Subsystem

When in doubt concerning the choices during the Q&A process, user can ask the expert system "why" for explanation. For instance, when the system asks user security questions, user hits the "why" button and the systems responds: Overly low security level exposes the system to invasion; overly high security level sometimes prevents normal browsing of the Web page; the recommended level is medium or medium high. This way user understands the purpose of the system's question and learns about proper setup. When in doubt of the recommendations and answers inferred by the system, user can hit the "how" button and the system will display the inference process to help user understand the rule and basis of the inference.

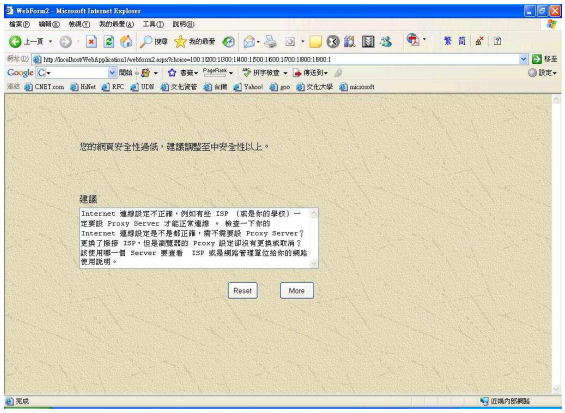

Figure 5 Explanation Screen

6 Integration of Expert Systems and Heterogeneous Systems

Currently, this study has successfully integrated Web Services with the expert system. Next we are going to utilize the cross-platform feature of Web Services to conduct the following experiments and propose three SOA-based integration models for expert systems.

(1) Building expert system knowledge bases (such as MySQL) of different platforms. Through the cross-platform communication feature of XML, the original system function is expanded. The original network problem diagnosis expert system (ES1+KB1) and computer hardware diagnosis knowledge base (KB2) are connected to integrate heterogeneous knowledge base platforms. The framework is shown in Figure 6.

Figure 6 SOA-Based Integration of Heterogeneous Knowledge Bases

(2) Utilizes the expert system shell of the JAVA-platform to develop a new expert system that integrates current .Net-platform expert system and JAVA-platform expert system in order to attain the goal of integration among heterogeneous program language platform systems as shown in Figure 7.

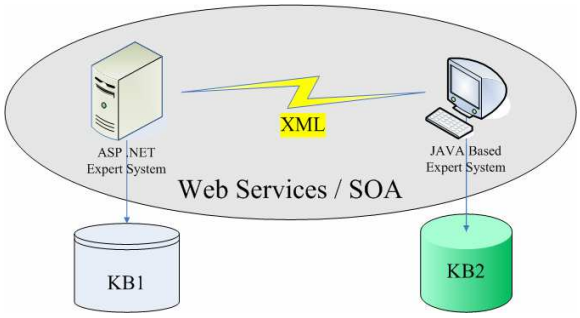

Figure 7 SOA-Based Integration of Heterogeneous Expert Systems

(3) Based on the loose coupling characteristic of Web Services and SOA, incorporating with different information systems, such as human resource management system, customer relationship management system, and procurement system, to enable the expert system to communicate with them and augment the service scope and function of the expert system as shown in Figure 8.

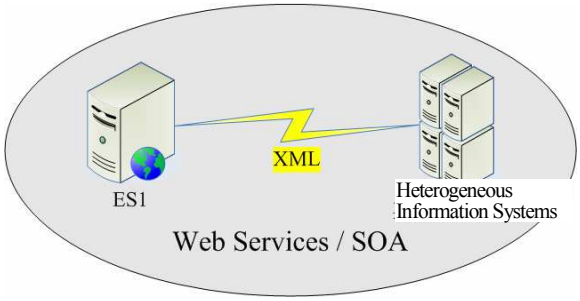

Figure 8 SOA-Based Integration of Expert System with other Heterogeneous Systems

The following table compares traditional Web-based expert systems with SOA-based expert systems.

Table 1 Comparison of Web-Based Expert Systems and SOA-Based Expert Systems

	Web-Based ES	SOA-Based ES
Cross platform	Inferior	Platform that supports different software/hardware and program languages
Software reuse	Need to know the details	Does not need to know the details
Expansion	Expansion requires additional programs	Have direct access to other services
Cost	More expensive	Less expensive
Security	Lower	Higher
Synchronism	Requires separate update	Will be updated when the original system /knowledge is updated

From the table 1 we can clearly see the edge of SOA-based expert systems. Besides the cross-platform feature, it is superior to Web-based expert system in software reusability and expansibility as explained below.

(1) Cross-platform feature: In comparison with traditional Web-based expert systems, of which most are applied to PC, server or PDA framework, SOA-based expert systems can support communication among various platforms, software/hardware and program languages.

(2) Software-reuse: When traditional Web-based system software is in reuse, we need to know details of the program or system while the reuse of SOA-based expert systems only requires definition of user interface and service content design.

(3) Expansibility: Adding any new functions or elements to traditional Web-based systems requires revision of the source code. Use of Web Services framework allows direct access current elements and services of other systems. The arrangement saves program development time.

(4) Cost: SOA-based expert systems save much program writing and system development time. It is both time and cost saving.

(5) Security: The database and system program codes of Web Services are normally distributed in different servers. In comparison with traditional Web-based systems, which place both the database and system in the same mainframe, it is safer. The risk has been distributed against damage.

(6) Synchronism: This is the most significant edge of using SOA-based expert systems. In the past, knowledge or system update of Web-based expert systems normally affected the entire structure and made

it very difficult. For SOA-based expert systems, their knowledge or system update requires only modification of the original program or its knowledge base. All other systems that refer to the original program will be updated automatically. To the expert system, this is very important. Expert knowledge acquisition and knowledge base establishment of traditional Web-based expert system is very involved. In addition to possession of expert knowledge and converting the knowledge into rule, frame and semantic networks, the acquired knowledge content has to be gradually established in different systems. The time and manpower it consumes often exceeds the benefit of the expert system itself. Yet through synchronous update of Web Services, system maintenance staffs only need to change the original program – the rest of the knowledge and system contents of the expert system will be updated synchronously.

7 Conclusion

This study incorporates SOA concept and expert system technology and develops SOA-based expert systems primarily for diagnosis of computer network problems. The purpose is to help users resolve Web-related problems such as inability to log in, abnormally slow Internet browsing speed, or frequently disconnection. Through the special Q&A series of the expert system, users receive a set of recommended solutions, get to understand their network problems and pinpoint the keys to solving their problems.

Further, the features of Web Services enable this network problem diagnosis expert system to provide more service platforms than those by traditional Web-based expert systems. It's available to users whether they are using PC, PDA, cellular phone, Windows, Macintosh, Linux or any other HTTP platform. More important is the fact that when the system is completed, all the Web service providers can retrieve the system freely and efficiently as an element. This way, they will be able to furnish more user services and help users to resolve network problems and enhance their knowledge.

In the area of future development, we propose integration of SOA-based expert system with other heterogeneous systems to provide a speedy, convenient, low-cost and cross-platform solution for development and application of the expert system. In the future, following registration of service elements with UDDI, any expert system will be able to join other systems, provide more diversified services and take the application and development of expert systems to a new high.

References:

[1] B. H. Shen, "Smart Client Development - Using Web Services & SOA," <http://www.microsoft.com/taiwan/msdn/columns/mvp/200412_smartclient.htm>, 2005.

[2] C. C. Chang, *Lecture Notes in Expert Systems and Knowledge Base*, Chinese Culture University, 2005.

[3] C. J. Jhang, *Applications that Utilize Grid Computing Concept to Enhance the Quality of Service-Oriented Systems*, Master's Thesis, Department of Management Information, Chung Yuan Christian University, 2003.

[4] E. Newcomer, translated by Y. K. Huang, *Understanding Web Services*, Pearson Education, Taipei, 2002.

[5] H. C. Jian, "Application of Service-Oriented Architecture," <http://www.microsoft.com/taiwan/msdn/columns/soa/SOA_overview_2004112901.htm>, 2004.

[6] P. Baglietto, M. Maresca, A. Parodi, and N. Zingirian, "Stepwise Deployment Methodology of Service Oriented Architecture for Business Communities", *Information and Software Technology*, 47(6), 2005, pp. 427-436.

[7] BEA, "Technical White Paper：IT Transformation to Service-Oriented Architecture", <http://www.bea.com/framework.jsp?CNT=wp_abst00027.htm&FP=/content/news_events/white_papers>, 2004.

[8] P. Gottschalk, "Expert System at Stage IV of the Knowledge Management Technology Stage Model: The Case of Police Investigations", *Expert Systems with Applications*, 31(3), 2006, pp. 617-628.

[9] W. Kim, Y. U. Song, and J. S. Hong, "Web Enabled Expert Systems Using Hyperlink-Based Interface", *Expert Systems with Applications*, 28(1), 2005, pp. 79-91.

[10] A. V. S. Rao and D. K. Pratihar, "Fuzzy Logic-Based Expert System to Predict the Results of Finite Element Analysis", *Knowledge-Based Systems*, 20(1), 2007, pp. 37-50.

[11] B. Tomic, J. Jovanovic, and V. Devedzic, "JavaDON: An Open-Source Expert System Shell", *Expert Systems with Applications*, 31(3), 2006, pp.595-606.

[12] M. Younas, I. Awan, and D. Duce, "An Efficient Composition of Web Services with Active Network Support", *Expert Systems with Applications*, 31(4), 2006, pp. 859-869.