Web Services: XML-based system integrated techniques

Shien-Chiang Yu and Ruey-Shun Chen

The authors

Shien-Chiang Yu is a Senior Manager, Systems Integration Department, Solomon Co., Taiwan.
Ruey-Shun Chen is Associate Professor at the Institute of Information Management, National Chiao-Tung University, Taiwan.

Keywords

Applications, Integration, Internet, Extensible Markup Language, Electronic data interchange

Abstract

The integration of information systems must consider various aspects, including the individuals of an organization, cooperating with other vendors, and the connections within this organization. The advantages of integration include shortening the negotiation process among the organization and the cooperating vendors, saving time for the users, and identifying the interface management. However, the greatest difficulties are how to integrate different system platforms and implement technical aspects into a suitable Web-interface for users to operate. Information technology companies have developed Web Services which rely on the flexibility of the Extended Mark-up Language. The services not only apply to integrated information systems, but also provide open services in Web environments. This article evaluates the feasibility of Web Services for application in integrated library information systems.

Electronic access

The Emerald Research Register for this journal is available at
http://www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
http://www.emeraldinsight.com/0264-0473.htm

The Electronic Library
Volume 21 · Number 4 · 2003 · pp. 358-366
 · ISSN 0264-0473
DOI 10.1108/02640470310491586

Introduction

Library organizations must continuously evolve in order to keep up with the changing information environment. The most common ways are to purchase or develop information systems to apply to these requirements when libraries face novel information applications. Because of developments in information technology, libraries must extend the scope of their automated services. After developing information systems to provide services, libraries will have many information systems for different demands, such as a bibliographic automation system, a current contents system, an online database, multimedia/metadata system, and so on. With the increasing use of the Internet, and the trend of integrated application, how to integrate the various systems in the library to provide a single operations interface for operators and users is a very important requirement.

However, due to the different sources and time of design, these library information systems, including the operation system, programming language, message standards, and even character-codes, may not be all uniform. Libraries thus face many difficulties when they attempt to integrate these systems. Generally, many libraries re-develop, purchase a new system, or extend system modules to replace existing systems in order to achieve integration. The major scope and variety of integrated modes include network integration, data consistence, transfer into open platforms, database integration, system architecture consolidation, etc. But many library organizations cannot integrate systems simply by purchasing a new system due to budget limitations, operating habits, or human resources, etc. The alternate method of indirectly achieving integration is by using a data import/export or application programming interface (API) to call remote systems. But such a method cannot immediately achieve the demands of the process of data import/export to transfer information between systems. In addition, systems will be more complex and unstable when using API to call other systems.

According to the goal of system integration mentioned above and based on the interoperability, scalability and flexibility of Extended Mark-up Language (XML), a novel Web Services model has been developed. These Web Services are not applied to

The Electronic Library
Volume 21 · Number 4 · 2003 · 358-366

integrate legacy systems, but rather they provide an open service in the Web world. This article discusses the feasibility of using Web Services, and which model to use for integration of library information systems.

Background

The integration of systems applications must be built on three bases: message communication, data format transparency, and business process automation. Business process automation is the major part which provides and connects the information flow among systems, and automates the process procedures. The message exchange and process part is used for integrated applications as a message hub for internal systems, and serves as a unified message channel for external systems. It also provides an exchange standard of information such as Dublin Core which is used in libraries/museums. It is necessary to use message service communication among systems, no matter what these systems are.

'... The user interface requires a material change in the process of application development. From text mode or window form to Web browser mode, the complexity and maintainability of the client must be simplified...'

Based on the language-neutral and platform-neutral of XML, automation systems can be readily modified for data exchange. Using XML as the major technique for integration of information systems provides the medium of information flow for message communication, data format, and business process automation. Enterprise application integration (EAI) is the base for message communication which uses XML to integrate legacy systems, regardless of these systems' programming language, operation platform or database. XML can be used to achieve the target of signal process portal over the Internet by the information exchange standard for message communication among systems, and also to envelop the remote process call (RPC) to require other systems or respond data for other systems' requirement.

In the application development layer in a library's information system, there should be three major phases, as listed below:

(1) *Initial stage*. Most systems are simple processing with bibliographic records in this phase.
(2) *Extension stage*. Here, most systems advance to database management and analysis, and extend to various information services like current content, full-text, multimedia, metadata, etc.
(3) *Mature period*. The service modes of information systems are oriented to personalization, and users can define service modes themselves.

The user interface requires a material change in the process of application development. From text mode or window form to Web browser mode, the complexity and maintainability of the client must be simplified. In addition, a library can handle more types of data by the presentation of conjoined multimedia. For these reasons, libraries must accommodate various user demands, information systems and operation platforms among inter-libraries.

The advantages of integration are that it can simplify the transaction flow, reduce process time and provide management consistency among inter-libraries or cooperative partners. However, how to integrate various different platforms or implement technical requirements into a single user interface is a very demanding issue. Therefore, using XML as the standard for object-envelope techniques enables data exchange across various platforms, operation systems and programming languages. A client uses the Simple Object Access Protocol (SOAP) which uses the XML standard to envelop and exchange information, and which implements the method of remote procedure call. Therefore, a client can easily call these functions by SOAP, regardless of what programming language is used by client or server to implement objects or Common Gateway Interface (CGI) application programs. This is the basic concept of Web Services. Regardless of the application, Web Services can provide for systems integration, giving flexible, loosely coupled integration that yields systems that can be decomposed and recomposed to reflect changes in the library.

The Electronic Library
Volume 21 · Number 4 · 2003 · 358-366

Type of integration

The problem of integrating information systems that are not only merged into whole systems of entire units in the specific organization of the library is very broad. Furthermore, consideration must also be given to the cooperative partners or other information systems in an organization, such as integrating a bibliographic automatic system with an enterprise resource planning (ERP) system used to acquire bibliographic/periodical information online, or integrating a work attendance system between the library and the personnel department.

To achieve this, we can refer to the methods and structures of integration which have been used in industry as references for the integrated modes in libraries. Application integration means creating or modifying the interactions among semi-autonomous but related application systems, including purchased packages, legacy applications and new Web Services. EAI entails integrating applications and enterprise data sources so that they can easily share business processes and data. EAI can be used to integrate information systems and according to the integrated mode, four types of data interchange can currently be distinguished: electronic data interchange (EDI), direct application integration, closed process integration, and open process integration.

EDI

Previously, EDI was the most popular integrated mode. In it, all of the data which were transferred and enveloped by X.12, EDIFACT, and the Z39.50 information access protocol belonged to the EDI (Bearman and Perkins, 1993). Nowadays, most information transmission modes include FTP, e-mail, specific e-mail, and socket mode, etc. But EDI must process over a value-added network (VAN) and while VAN does not have higher cost, it uses a closed method to transfer data, no matter what EDI mode is employed. This reduces the process effect and increases the implementation barrier, thus preventing small libraries adopting it. Because XML runs over the Internet, it eliminates fees to EDI value-added network providers and so helps to popularize integration.

Direct application integration

This mode is a common application method which is applied together with an EAI solution. It can integrate a system with other external application systems, integrating all of the correspondent suppliers (or third parties) in the application layer to come into a value-chain. However, this mode must interact with API, which belongs to the information system. Essentially, API provides a set of object-oriented interfaces which are opened within the intra-system and which are used to supply other system resources, such as application logic, middle-ware, database, etc. Other systems make a function call by passing on requirements and getting responses from the remote system's API. An integration-broker or build-in adapter, which is normally provided by the software developer among systems when using this integration mode, is used in order to connect with API and handle the data interchange. Systems can be integrated under an integration-broker to achieve the goal of data interchanges.

Closed process integration

Actually, inter-library systems do not only consider data interchange. The interactive relationship must consider cooperation among libraries, including borrower privileges, charge type, service scope, etc. These interactive and composed orders are called business processing. Business processing integration must rely on the co-processing flow to define order, organized architecture, event, process logic, and message flow among inter-libraries. Business processing promotes integration for an interactive relationship to enhance the services to users. In closed business processing, even though the process flow is already integrated, there still can not be full transparency or active management flow among these library systems. Because there is just reply data to the other sides depending on demand, this is a kind of closed process integration.

Open process integration

Open process integration provides the concept of process-sharing, taking processes from information systems into peer-to-peer relationships (which is called peer level). In this mode, every component of the information system can handle inner-

The Electronic Library
Volume 21 · Number 4 · 2003 · 358-366

processes by itself, but still can be integrated with other components. The information system can decide which events related with the process should be put into a co-process to be shared with other information systems, or decide which events will be processed only internally.

Web Services integration

What are Web Services?

The design point of traditional software engineering is construction and model design. From model design to object-oriented development the software engineer introduces the inheritance, encapsulation and polymorphism techniques. System analysis and development also function this way. The concept of system analysis and development join with SOAP message encapsulation, which is based on XML to derive the applied mechanism of Web Services. The World Wide Web consortium (W3C) defines a Web Service as follows:

> A Web service is a software application identified by a URI, 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 (Austin *et al.*, 2002).

On the other hand, IBM offers this definition:

> A Web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging. A Web service is described using a standard, formal XML notion, called its service description. It covers all the details necessary to interact with the service, including message formats (that detail the operations), transport protocols and location. The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written. This allows and encourages Web Services-based applications to be loosely coupled, component-oriented, cross-technology implementations. Web Services fulfill a specific task or a set of tasks. They can be used alone or with other Web Services to carry out a complex aggregation or a business transaction (Kreger, 2001).

Microsoft has two definitions for Web Services:

> A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web Service is implemented (MSDN Library, 2002).

The other Microsoft definition is:

> A Web service is programmable application logic accessible using standard Internet protocols. Web services combine the best aspects of component-based development and the Web. Like components, Web services represent black-box functionality that can be reused without worrying about how the service is implemented. Unlike current component technologies, Web services are not accessed via object-model-specific protocols, such as the distributed Component Object Model (DCOM), Remote Method Invocation (RMI), or Internet Inter-ORB Protocol (IIOP). Instead, Web services are accessed via ubiquitous Web protocols and data formats, such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). Furthermore, a Web Service interface is defined strictly in terms of the messages the Web Service accepts and generates. Consumers of the Web service can be implemented on any platform in any programming language, as long as they can create and consume the messages defined for the Web Service interface (Kirtland, 2001).

From these definitions one important point is that Web Services do not necessarily exist on the Web, they can exist anywhere on the network, Internet or intranet. Some Web Services can be invoked by a simple method invocation in the same operating system process, or perhaps using shared memory between tightly coupled processes running on the same machine. In fact, Web Services have little relationship with the browser-centric, HTML-focused Web (Graham *et al.*, 2001).

From the perspective of the overall application, the major Web Services are applied to system integration. This includes integrated information systems both intra- or extra-organization.

'... Some Web Services can be invoked by a simple method invocation in the same operating system process, or perhaps using shared memory between tightly coupled processes running on the same machine...'

An information system supplies Web Services to allow other systems to archive the integrated target, no matter what different hardware platforms or systems are developed using different languages. Utilizing SOAP to

The Electronic Library
Volume 21 · Number 4 · 2003 · 358-366

communicate messages simplifies the processes and results. In addition, the traditional peer-to-peer integrated mode can be improved, using Web Services mode to achieve one-to-multiply integration.

Figure 1 shows the principle of the SOAP and Web Services process on the Web. Clients parse the required information into XML format and encapsulate it to SOAP objects. They then utilize the Web to transfer it to a server site, and then use the service interface of Web Services to call an inner-function which is a process in the server. When the process of this function is finished, the server will parse the result into XML format and transfer it back to the client. Because SOAP is implemented over HTTP, therefore a SOAP object can call any platform which supports HTTP, and the SOAP object can pass through firewalls to solve tasks which many software applications are unable to perform.

Basic techniques

The concept of Web Services is very simple. Most of the basic techniques used are open standards which are popularly accepted. Software developers can utilize existing standards adding in original software language or hardware platforms when implementing the techniques of Web Services.

Web Services dynamically issue, search and process services using the Internet. However, the overall technical standard has not yet been established. The major mature standards include the Web Services Description Language (WSDL), SOAP, Universal Description, Discovery and Integration (UDDI) (Seely, 2002). The relationship of the entire process is shown as Figure 2.

- *HTTP*. The main technology currently used for SOAP transport is the Hypertext Transfer Protocol (HTTP), but SOAP specification does not require HTTP. Although it might be able to use SMTP, FTP or other methods, HTTP is currently the most common.
- *SOAP*. The SOAP is a lightweight and simple XML-based protocol that is designed to exchange structured and typed information on the Web (Box *et al.*, 2000). The purpose of SOAP is to enable rich and automatic Web Services based on a shared and open Web infrastructure. SOAP can be used in combination with a variety of existing Internet protocols and formats including HTTP, SMTP, and MIME and can support a wide range of applications from messaging systems to RPC (Graham *et al.*, 2001).
- *WSDL*. Web Services Description Language (WSDL) offers an XML protocol for describing network services based on XML. WSDL is currently also submitted to the W3C as a note, and steps are underway to work formally on WSDL in the context of related work within the W3C (Christensen *et al.*, 2001). WSDL uses SOAP for its messaging functions to exchange data among remote applications, but it can also use MIME and native HTTP transport protocols (Ogbuji, 2000).
- *UDDI*. The Universal Description, Discovery and Integration (UDDI) project is an industry initiative that is working to enable businesses to quickly, easily, and dynamically find and transact with one another. UDDI can benefit businesses of all sizes by creating a global, platform-independent, open architecture for describing businesses and services, discovering those businesses and services, and integrating businesses using the Internet. Any kind of service can be registered in the UDDI business registry, but the primary intent behind UDDI is to provide a global registry for Web Services (Web Services Mall, n.d.).

Figure 1 Browser passes the request to the SOAP proxy, which then invokes the code behind the Web Service

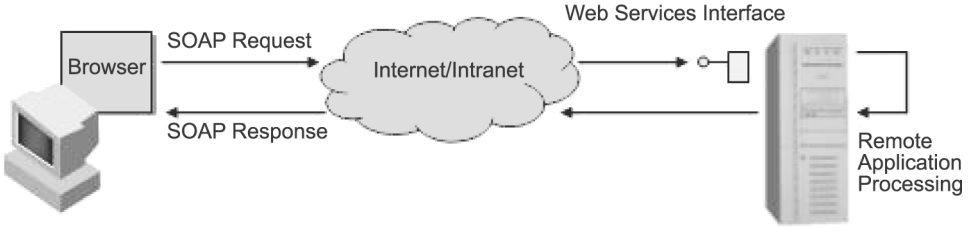
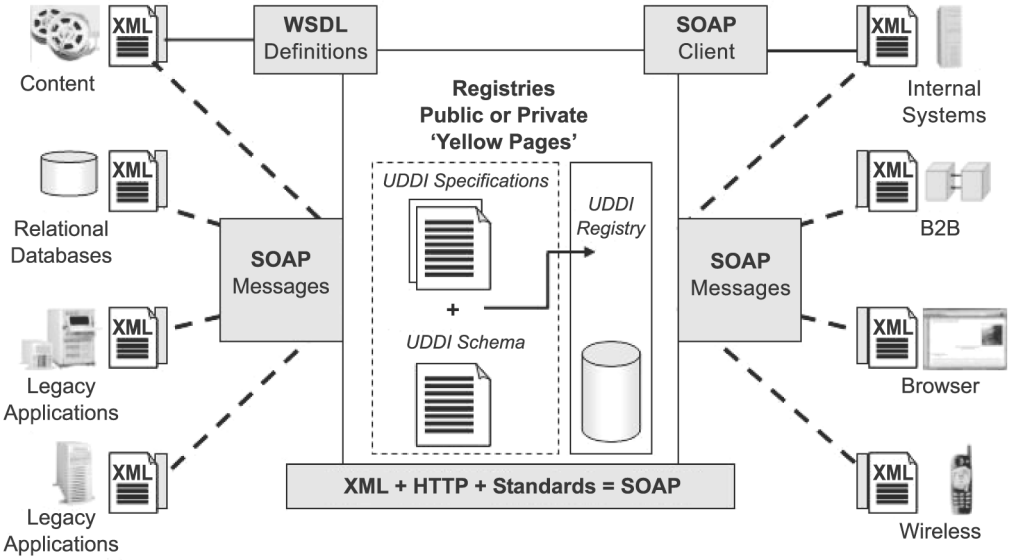


Figure 2 Providing Web services using XML, SOAP, WSDL and UDDI



Source: Meloan (n.d.)

Characteristics and advantage of integration

To meet the above requirements of the application integration among systems, Web Services have several characteristics:

- they can be combined with messages;
- they can search by information of application;
- they offer low cost and simple integration with legacy application;
- they provide the connection window for cooperative partners; and
- they permit full-scale integration.

System can search valuable Web Services and process the information request/response among the interfaces of Web Services from the Internet through UDDI. For example, the purpose of any library is to obtain, preserve, and make available recorded knowledge (Katz, 1992). But now, there is so much knowledge that cannot be collected due to the rapid expansion of the Internet and the availability of e-content. Although users can use many open distributed protocols such as Z39.50 or the Open Archives Initiative (OAI) to query or retrieve documents from service providers, the majority of these protocols are based on established relationships between data provider and service provider. Thus, both sides must set up appropriate service parameters or lists, and the major scope of service is to document or catalog delivery. By applying Web Services and UDDI, a library's system can actively provide interfaces or search many favorable services for users from the Internet, and develop interaction automatically among systems or users.

From the perspective of the overall application, applying Web Services provides six main advantages:

(1) *Simplicity*. The popular integrated mode, which has been adopted by systems in the past, includes Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA), or using data exchange mode of EDI to achieve integration. These modes will be complex and cannot definitely be implemented in a legacy system. In contrast, when deploying the framework of Web Services in a legacy system, then it is easy to construct novel business processing which can cross multi-application systems. Thus, Web Services simplify the design, development, maintenance and usage more than other modes.

(2) *Open standard*. Compared with the traditional mode of system integration, existing protocols now adopted by Web Services include UDDI, SOAP, WSDL, and HTTP. Thus, enterprises or libraries do not need to invest more to support other communication protocols.

(3) *Flexibility*. Point-to-point communication is the major method used to integrate systems in the connection mode. This mode will enclose each connective

The Electronic Library
Volume 21 · Number 4 · 2003 · 358-366

system, and modifying one of these systems will influence the others. In contrast, the integrated mode which is utilized by Web Services uses announcement services to integrate other systems, keeping the flexibility between systems.

(4) *Broad scope*. In traditional systems integration, one application system provides only one integrated portal interface, but Web Services divide a system into multiple logical and independent units. Web Services also provide multiple connect points (interfaces) for other systems to integrate with. For example, an order system can use Web Services to provide orders received, order status query, order confirmation, collection, payment, etc. in one single integrated protocol.

(5) *Efficiency*. Web Services can provide the advantage of multiple entrances integrated into a single portal. This implies that the integrated mode of Web Services is more efficient than traditional mode.

(6) *More dynamic*. If the Web Services mode is adopted under EAI processing, only appropriate business services which are published by "Services providers" can combine with business processing to achieve the integrative target. Compared with traditional integrative modes, Web Services provide stability using a dynamic integrated interface to integrate systems.

Issues and limitations

The service mechanism of Web Services is based on interaction of SOAP components. Thus Web Services will face many issues which already exist in the network environment, such as privilege, tracing the status of service, service log, charge mode, and so on.

Reliability

Web Services uses inter-operation of objects to achieve service. The major actions in this situation include transferring, manipulating, and recording information. After calling, the remote service will receive one response value, but the system cannot acquire the process of how it works in the remote service. For example, if one Web site declares that it has provided the service of selective dissemination of information (SDI), users register their e-mail addresses only once, and then apply to the services for bibliographies and periodical contents from many libraries. If someone builds this SDI service, a major objective may be to collect each user's profile and sell them outside, which would cause serious damage to the library and users. There is still not any trustworthy mechanism to avoid in the whole process.

Security

Web Services still face the security issue just as on the Internet network. It cannot be ensured that hackers will not retrieve information in the process of information transfer. For service after the WSDL, all of the parameters and response values will be in a plain code, which would be easier for a hacker to get or analyze information, so information security must be considered. The key point of Web Services is communication between system and service, with issues of identification not only directly to individuals, but also to verify service objects. However, verifying service is more difficult than verifying individuals.

Transaction

The issue of transaction is actually the greatest barrier for Web Services application. In the construction of Web Services, data could be stored, managed and transferred to distributed locations. Thus, when one transaction is triggered, it will generate a distributed transaction, and distributed transactions have two basic problems. The primary one is the issue of transaction time since one transaction must be completed before timeout. But because Web Services are distributed over the Internet, the response time of a transaction will be restricted by the bandwidth or load of the remote server. Shortening the timeout setting parameter will raise the fault rate of transactions, whereas expanding it will affect the quality of service.

The second problem is transaction rollback. When one transaction fails, the system must rollback all changes in this transaction, i.e. return these changes to the original status. Traditionally, centralized transactions handle all of the resources including databases and tables. When one transaction starts, the major method is data-lock to prevent other transactions from changing the same record at

The Electronic Library
Volume 21 · Number 4 · 2003 · 358-366

the same time. With Web Services, using data-lock will delay the response time, but not using data-lock will cause the problem of data synchronized change.

Protocol limitations

(1) *HTTP*. HTTP has two major issues when translating packets:
- It does not guarantee that the packet will translate to other side successfully.
- It does not guarantee that the packet will translate to other side by order.

When network bandwidth is congested, HTTP packets may be discarded, so the bandwidth will be a network bottleneck. Insufficient bandwidth causes delay of message delivery and the issue of message synchronization affects the executing efficiency of Web Services.

(2) *SOAP*. Essentially, SOAP is the major processing protocol in Web Services, that is to say, Web Services utilize SOAP to process the service. The efficiency of SOAP will directly affect the efficiency of Web Services. Because SOAP encapsulates messages using the XML standard, it takes some time when the system fetches the SOAP envelope from the SOAP message, or parses the XML document from the SOAP envelope. Especially, most parsers will process data type checking and verify that they are well-formed and validated. Thus, the program size, execution time and memory requirement of the parser will consume a large amount of system resources.

Using XML requires more space to store data than using text mode, and a larger size will reduce the efficiency through the Internet to translate messages. Ways in which to shorten the SOAP message must be considered when designing Web Services. In addition, although many international organizations declare that they will support SOAP, it still has many undefined standards in how to declare and explain the message schema to present information, and how to require and respond conveniently between systems.

(3) *UDDI*. The main function of UDDI is providing Web Services that can search all of the necessary services from UDDI. Based on this target, UDDI does not need more effect on demand, but uses a dynamic method to search suitable services to provide Web Services. As a result the function and stability of the UDDI search engine will be a major issue for expanding Web Services.

Conclusion

In the beginning a novel technique is usually accompanied by many related aspects, include technical theme aspects, application aspects, logistic aspects, etc. Although Web Services have advantages over traditional modes of system integration, currently they can only provide functional integration, i.e. just the basic "request/response" function, rather than total transactions. Thus, Web Services cannot yet replace traditional integrated modes, but only provide another option for an integrated technique. After the Web Services structure is more mature, information systems will tend to have a service-oriented mode. Eventually, library information systems will not be limited to bibliographic operations, and will be able to automatically search and connect to suitable Web Services on the Internet to provide a more extensive solution for library service and knowledge management.

References

Austin, D., Barbir, A. and Garg, S. (2002), "Web Services architecture requirements", *W3C*, available at: www.w3.org/TR/2002/WD-wsa-reqs-20020429

Bearman, D. and Perkins, J. (1993), "Standards framework for the computer interchange of museum information", *CIMI*, available at: www.cni.org/pub/CIMI/part4.html

Box, D., Ehrebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, A.F., Thatte, S. and Winer, D. (2000), "Simple Object Access Protocol (SOAP) 1.1", available at: www.w3.org/TR/2000/NOTE-SOAP-20000508/

Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. (2001), "Web Services Description Language (WSDL) 1.1", available at: www.w3.org/TR/wsdl

Graham, S., Simeomov, S., Boubez, T., Daniels, G., Davis, D., Nakamura, Y. and Neyama, R. (2001), *Building Web Services with Java*, SAMS, Indiana.

Katz, W.A. (1992), *Introduction to Reference Work: Volume I*, 6th ed., McGraw-Hill, New York, NY, p. 10.

Kirtland, M. (2001), "A platform for Web Services", available at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnWebsrv/html/Websvcs_platform.asp

Kreger, H. (2001), "Web Services Conceptual Architecture (WSCA 1.0)", available at: www-3.ibm.com/software/solutions/Webservices/pdf/WSCA.pdf

Meloan, (n.d.), "JavaTM2, enterprise edition and .Net: can we live together?", available at http://java.sun.com/features/2002/04/j2eenms.html

MSDN Library (2002), "XML Web Services", available at: http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000442

Ogbuji, U. (2000), "Using WSDL in SOAP applications: an introduction to WSDL for SOAP programmers", *IBM Developer Works*, available at: www-106.ibm.com/developerworks/library/ws-soap/index.html

Seely, S. (2002), SOAP: *Cross Platform Web Service Development Using XML*, Prentice-Hall, Englewood Cliffs, NJ, p. 163.

Web Services Mall (n.d.), "Universal Description Discovery Integration (UDDI) standard", available at: www.Webservicesmall.com/profile.asp?pid=105&ptitle=Universal%20Description%20Discovery%20Integration%20(UDDI)%20standard

About the authors

Shien-Chiang Yu is a PhD Student in the Institute of Information Management, National Chiao-Tung University in Taiwan. He received his MS degree from the Institute of Library and Information Science, Fu-Jen Catholic University in1997. From 1998-2001 he was Project Manager in the Library Information Department of TOP Business Machine Company, Taiwan where he worked with the SPYDUS Library System. Currently he is Senior Manager in the Systems Integration Department of Solomon Co. Ltd where his major tasks are the development of smart cards and knowledge management systems. His research interests include spatial database, electronic commerce, information retrieval, and metadata. He can be contacted at: u8834804@cc.nctu.edu.tw

Ruey-Shun Chen obtained his PhD from the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan in 1995. Since that time he has been Associate Professor in the Institute of Information Management, National Chiao-Tung University. His research interests include reliability and performance evaluation, networks, information management, distributed systems, and digital libraries. He can be contacted at: rschen@bis03.iim.nctu.edu.tw