

Association for Information Systems AIS Electronic Library (AISeL)

PACIS 2000 Proceedings

Pacific Asia Conference on Information Systems
(PACIS)

December 2000

An Implementation of a Web-based Timetabling System

Shu Wing Chan
Up'n'Away Net Solutions

Weigang Zhao
Curtin University of Technology

Follow this and additional works at: <http://aisel.aisnet.org/pacis2000>

Recommended Citation

Chan, Shu Wing and Zhao, Weigang, "An Implementation of a Web-based Timetabling System" (2000). *PACIS 2000 Proceedings*. 1.
<http://aisel.aisnet.org/pacis2000/1>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

An Implementation of A Web-based Timetabling System

Shu Wing CHAN
Up'n'Away Net Solutions
P.O. Box 88 Floreat Forum
WA 6014 Australia
+61 8 9367 9088
chansw@upnaway.com

Weigang ZHAO
School of Information Systems
Curtin University of Technology
GPO Box U1987 Perth WA 6845 Australia
+61 8 9266 7642
zhaoc@cbs.curtin.edu.au

Abstract

This paper first describes various interfacing mechanisms between the Web and databases. Secondly, features of various web-database interfacing mechanisms are discussed. Thirdly, a web-based timetabling system implemented using one of the mechanisms is described. Finally, conclusions and future work are reported.

Keywords: Web-based information system, Web database, Web-based applications

1. Introduction

The World Wide Web (known as “WWW” or “Web”) is growing at a phenomenal rate. The current Web is largely based on file system technology, which can deal well with the resources that are primarily static. However, with the unprecedented growth of resources, it is no longer adequate to rely on this conventional file technology for organising, storing and accessing large amount of information on the Web. Thus, many large Web sites today are turning to database technology to keep track of the increasing amount of data. Database technology has played a critical role in the information management field during the past years. It is believed that the integration of the Web and database technology will bring many opportunities for creating advanced information management applications (Feng, 1998).

With the increasing popularity and advancement of the Web technology, many legacy information and database systems are being migrated to the Internet and the Web environments. Many innovative web-based applications have been developed. Web-to-database tools have been made available for developing such applications. With live Web databases, travel agents can keep flight schedules and fares updated; businesses can update inventory lists and prices; customers can look up the latest prices and order products online; employee training can be an interactive experience; and a much greater range of information publishing applications is possible.

Taking simple data from a database and placing it on the Web is a relatively simple task. However, in most cases, the corporate data is maintained in a variety of sources, including legacy, relational, and object databases. It is much more complicated when these diverse data sources must be queried or updated (Carriere, 1997). With the level of complexity of the applications increases, there is a growing concern about the way that these applications were developed. “Web Engineering”, as a new discipline, has been coined and promoted by a series of workshops and seminars (Murugesan, 1999).

There are many players in the industry taking this challenge. These include major database vendors, mainframe vendors, third party software firms, Web browser vendors, and Web server vendors. A wide range of tools and philosophies has been proposed for connecting and integrating the Web and databases (Kim, 1997; Ashenfelter, 1999). In last paper (Zhao, 1999), we presented a web-based configuration database application implemented using different web-to-database interfacing techniques. This paper is to describe a real-life web-based timetabling system developed and deployed at Curtin Business School, Curtin University of Technology.

The remainder of the paper is organised in three sections. Section 2 describes various web-to-database connection approaches and presents features of different approaches. Functions and operations of the web-based timetabling system are described in Section 3. Conclusions and future work are reported in Section 4.

2. Web-To-Database Connection Approaches

Delivering data over the Web is cost effective and fast, and gives Internet users easy access to databases from any locations. Users hope to access databases via Web browsers with the same functions as provided by normal database application software. Businesses want to provide their users or customers various functions such as purchasing goods, tracking orders, searching through catalogues, receiving customised content, and viewing interesting graphics. The Web-to-database integration has become central to the jobs of corporate information systems construction.

Making database information available to Web users requires converting it from the database format to a markup language such as HTML or XML. Database packages store information in files optimized for quick access by front-end programs. When a Web server sends information to a client, the internal database format must be converted to HTML so that it is displayed correctly (Reichard, 1996).

To build a bridge between Web and enterprise database, a number of alternative technologies and architectures have been available. These include:

- CGI (Common Gateway Interface) is a Web standard for accessing external programs, to integrate databases with Web servers. The CGI dynamically generates HTML documents from back-end databases;
- Web server APIs, such as Microsoft's Information Server API (ISAPI), Netscape API (NSAPI), are invoked by third party software to access remote databases;
- Web-ODBC (Open Database Connectivity) gateways rely on an open API (Application Programming Interface) to access to database systems;
- Vendor-specific Web browser/data warehousing interfaces are in response to the inherent advantages of the two technologies;

- JDBC (Java Database Connectivity) is used in its Java programming language to program Java applets to access back-end databases.

Each of the above technologies has strengths and weaknesses. Several factors should be considered when making selections. These include the complexity of data, the speed of deployment, the expected number of simultaneous users, and the frequency of database updates. However, new technology is emerging and several tools are already available that make this Web-to-database access optimised for improved performance (Carriere, 1997).

2.1 CGI

Common Gateway Interface (CGI) is the oldest and the most commonly used method for implementing a Web database gateway. The CGI scripts/programs can be written in a number of programming languages, which include Perl, C/C++, TCL, Visual Basic etc. CGI is the standard feature on all web servers across different platforms and it does not require any special settings on the web server and web browser. In CGI, an HTML form is used as the graphical user interface for obtaining user requests. The server then transfers the request to the gateway program. The gateway program either processes the request itself or passes the request to another program/system to process. The program/system generates the results into HTML format and transfers back to the web server. The web server presents the result back to the client.

Currently, Perl is the dominant language for writing CGI applications. Perl itself does not have any database support but it has a large range of modules providing database support. Two of the widely used database modules are DBI (DataBase Interface) and Win32 ODBC. These modules provide the unique API interface to access different vendor databases.

The advantage of CGI approach is it is simple to implement and freely available on all web servers on different platforms. However, it has several limitations. A process is spawned with each request and that is time-consuming and expensive in system resources. Moreover, it cannot allow any database connection to be maintained and that means each time a CGI script queries the database, a new connection is started between the CGI and the DBMS.

2.2 Active Server Page (ASP)

Active Sever Page (ASP) is a server-side scripting technology developed by Microsoft. It allows scripts to be embedded within the HTML pages in any scripting language. Currently, you can use VBScript, Jscript (Javascript from Microsoft) and PerlScript. It is shipped with Microsoft Internet Information Server (IIS) and O'Reilly Web Server Pro that run on Windows 95/98/NT platforms. With some third-party plugins like Chilli ASP, ASP can be run on other web servers and platforms.

Database connectivity is provided in ASP with any ODBC-compliant databases. With HTML form as the front-end, client-server applications can be built using ASP. It supports the execution of SQL statements and stored procedures.

Active Server Page executes faster than traditional CGI because it is running in the same process space as the web server. There is no need to create a separate process. Thus, it uses less memory and system resources. However, it can crash the web server when there is an error in the script because it is running in the same process space as the web server (Wille, 1997).

2.3 Java Servlet

A Java servlet is a generic server extension that expands the functionality of a server. A Java Servlet can be used to replace CGI. It runs inside a Java Virtual Machine (JVM) on the server so it is secure and portable. The difference between Java Servlet and Java Applet is Servlet does not require any Java support on the client as all the processing is carried out on the server side. Since Servlets are written in Java, they have full access to Java APIs and third-party component classes. Also, Java Servlets are portable across operating systems and web servers. Java Servlets are more efficient than CGI because they execute within the web server's process space and they persist between consecutive invocations.

A database gateway can be written in Java Servlets with JDBC. JDBC is a Java SQL API for accessing databases. It provides a set of classes and interfaces for writing database applications using a pure Java API. Like ODBC, JDBC provides a unique interface for virtually any relational databases. JDBC is built on ODBC and both of them are based on X/Open SQL CLI (Call Level Interface) except JDBC builds on the styles and virtues of Java (Hamilton, 1997; Hunter and Crawford, 1998).

2.4 PHP

PHP stands for Hypertext Preprocessor, which is a HTML embedding scripting language. Its syntax is similar to C and Perl and is freely available on Windows and Unix platforms. With a set of modules providing different database systems, PHP has been widely used on many database-driven web sites.

PHP can run as a CGI wrapper or in module mode. When it is running as a CGI wrapper, it cannot gain the advantage of persistent database connection due to the life cycle of a CGI process. The current popular method is to run PHP as a module (`mod_php`) that provides persistent database connection. (PHP)

2.5 mod_perl

`mod_perl` provides a Perl interface to Apache Web Server's C language API. `mod_perl` is linked and compiled together with Apache and Perl. This means a Perl interpreter is compiled with the Apache server. The main advantage of using `mod_perl` is its speed and power. Unlike CGI, it does not require a separate process each time when a request is made and the time to start a new Perl interpreter is eliminated. Also, you have full access to the web server functions that you can do a lot of things you cannot do with traditional CGI. The web-based database system can gain the advantage of persistent database connection when using `mod_perl`. However, with `mod_perl`, the developer has to make sure this is done properly as the same script will be continuously used when new requests come in. `mod_perl` is available on both Windows and Unix platforms (Stein and MacEachern, 1999).

2.6 Server API

An alternative to modifying or extending the abilities of the server is to use its API. APIs allow the developer to modify the server's default behavior and give it new capabilities. In addition to addressing some of the drawbacks of CGI, the use of an API offers other features and benefits, such as the ability to share data and communications resources with a server, the ability to share function libraries, and additional capabilities in authentication and error

handling. Because an API application remains in memory between client requests, information about a client can be stored and used again when the client makes another request (Frey, 1996).

For example, NSAPI gives developers a finer level of control by allowing functions to work on various levels, such as forms processing, error handling, and logging. In addition, it allows developers to have full access to the server's internal data structures, functions, and I/O abstractions. The I/O abstractions include those necessary to perform secure data transfer between the browser and the server, and access various system facilities such as semaphores transparently.

Microsoft's ISAPI is a high performance interface to back end applications running on the web server. Based on its own DLL ensuring significant performance over CGI. ISAPI is easy to use, well documented, and does not require complex programming. This API is being endorsed by an increasing number of server developers, allowing application developers to write for a single specification and to deliver on multiple platforms (Duan, 1996).

2.7 ODBC and JDBC

ODBC and JDBC are types of database access middleware. ODBC is, by far, the most popular database access middleware in use today. Vendor support for ODBC is pervasive. JDBC support isn't quite at the level of ODBC support, but JDBC is growing and flourishing. Database vendors and several third-party software houses offer ODBC and JDBC drivers for a variety of databases and operating environments.

From a network administrator's point of view, they consist of client and server driver software (i.e., program files). From a programmer's point of view, they are APIs that the programmer inserts in his or her software to store and retrieve database content. While a system analyst perceives ODBC or JDBC as a conceptual connection between the application and the database, database vendors regard ODBC and JDBC as ways to entice customers who say they want to use industry standard interfaces rather than proprietary ones. And managers of data processing department view ODBC and JDBC as insurance interfaces that offer managers some measure of flexibility should they find it necessary to replace one database product with another (Wong, 1997).

ODBC technology now allows Web servers to be used to directly connect with databases, rather than using third party solutions. JDBC can also directly access server ODBC drivers through a JDBC/ODBC Bridge driver, available from SunSoft. ODBC driver vendors are also building bridges from ODBC to JDBC. JDBC is intended for developing client/server applications to access a wide range of backend database resources.

3. Functions of A Web-based Timetabling System

This web-based online timetabling system was initially developed by School of Computing and has been enhanced and modified for Curtin Business School. The system comprises of a set of CGI scripts written in Perl 5 with OraPerl. Form validation and some user interface enhancements are performed with Javascript. Two sets of functions have been provided in this system. One set of functions is used by students and another set of functions is used by timetable administrators. The system runs on a Sun workstation with Apache 1.3.9 and Oracle8 Enterprise Edition Release 8.0.5.0.0 – Production.

3.1 Functional Components Used by Students

The student is welcome by the login page where the username, password and email address are needed for accessing the system. Any missing input will be prompted for.

The edit class page comes up next where the student enters up to 5 unit index numbers. Again, all the inputs are validated with JavaScript before they are sent to the server for further processing. On this page, the student can view what they have enrolled so far and they can remove an enrolled class or print out the timetable. It will also check the total number of units that each student is permitted to enroll does not exceed 5.

The available classes for each unit will bring up for the student to select. If there are no more class seats available for a unit, the student will be put into the waiting list for that unit automatically. Once the classes are selected, it will check if the unit(s) is already enrolled. Appropriate error message will be given. If everything is OK, the student can click the button to enroll the units and their details will be recorded into the database.

The functions of each script are described as follows:

Login.cgi

This script is used to authenticate students when they enter the system. Three user inputs are required. They are the student number (which acts as the username), the date of birth (which acts as the password) and the email address. After a student fills out all the required fields, he/she can click on the login button. If any of the input is missing, the student will be prompted with a dialog box (which is implemented using JavaScript). This script opens a database connection and retrieves the student number and date of birth and compares with the user input. It will also update the email address if it is not there or changed when the username and password are valid. Figure 1 shows the login page of the timetabling system used by students.




Figure 1. The login page of the timetabling system used by students.

Edit.cgi

This script presents the main page where the student can manipulate on his/her enrollment. If the student has already enrolled at least one unit, the timetable will be displayed on the top. At the same time, there is a hyperlink to print.cgi where the timetable is displayed on a separate page and can be printed out. On the left side of each enrolled unit, the student can click on the hyperlink 'delete' which calls up the remove.cgi to remove the enrolled unit. If the student has not enrolled anything yet, he/she can fill in the five boxes with the units he/she would like to enroll. There is an option to display a box with all the unit information for reference.

Remove.cgi

This script removes the selected unit from the appropriate tables in the database.

Print.cgi

The script prints the student timetable in a HTML table format on a separate page.

Student: 996760C

Day	Time	Unit Name	Room	Notes	Type
Monday	0900 - 1000	Intro To Program Concepts 102 (13601)	401.378	-	Tutorial
Monday	1000 - 1200	Intro To Program Concepts 102 (13601)	402.223	-	Lab
Monday	1600 - 1700	Intro To Program Concepts 102 (13601)	401.002	-	Lecture
Wednesday	0800 - 1000	Electronic Commerce (Iff) 201 (12031)	402.214	-	Lab
Wednesday	1000 - 1100	Economics (Macro) 202 (9792)	402.345	-	Tutorial
Wednesday	1600 - 1800	Economics (Macro) 202 (9792)	401.001	-	Lecture
Friday	0800 - 1000	Electronic Commerce (Iff) 201 (12031)	210.101	-	Lecture

Click [here](#) to return to previous Page

Figure 2. A student timetable

Times.cgi

This script gets the student number along with the units selected to enroll as input. The available classes for each selected unit will be printed in an option box. The student has to highlight the class they like to enroll and click on 'Continue'. If there is no class available for a unit, the student will be put on the waiting list and appropriate message is displayed on the screen.

Enrol.cgi

Enrol.cgi updates the student enrollment details to the database when the selected classes are still available. If the selected classes are not available at the time when the student is trying to submit, the student will be put on the waiting list automatically. After the updates on the

database, the student can print out the timetable or exit the system. Figure 2 displays a student timetable generated by the system.

3.2 Functional Components Used by Administrators

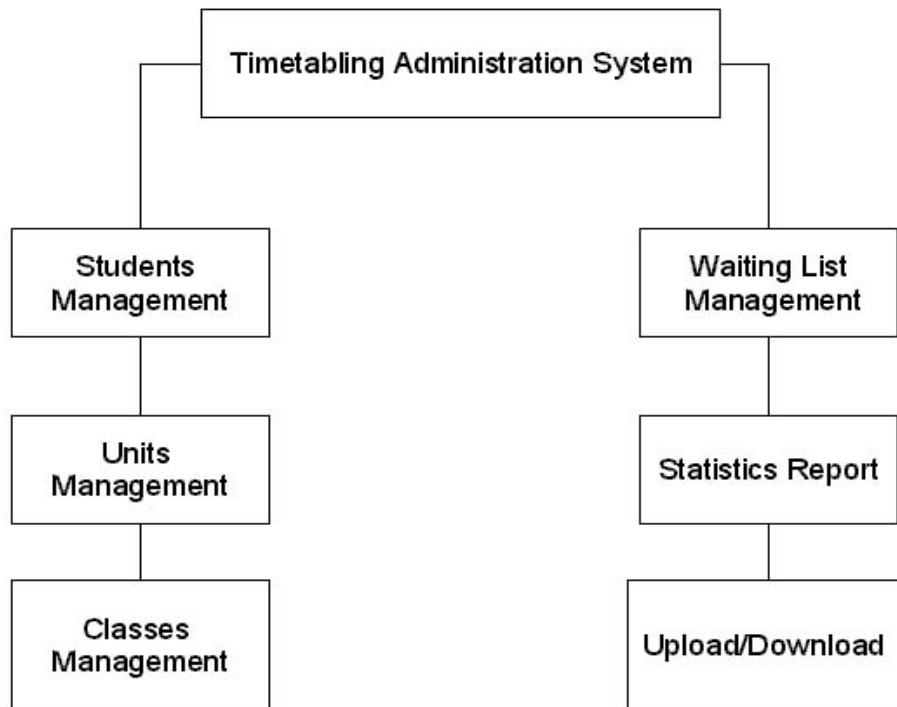


Figure 3. Functional components used by administrators

In the timetabling administration system, there are six major components as illustrated in Figure 3. These include students management, units management, classes management, waiting list management, statistics report and upload/download. The timetable administrator can manipulate the data in the database via the Web interface like adding classes, editing unit details, moving the students from the waitlist to the new classes and email to all the students for any changes. Also, the system allows the timetable administrator to upload large amount of username and password or class information via the Web interface and the insertion of data to the database is done automatically. Appropriate error message comes up if a problem is encountered. Moreover, the administrator can also download the classes and/or units enrolment information at anytime. With the use of HTML interface, that makes the person without any database and/or SQL knowledge to carry out the tasks as mentioned above. The functions of each component are described in the following sections.

3.2.1 Students Management

Add-student.cgi

This script allows the timetabling administrator to add a new student. The administrator inputs the name, student number, date of birth and email address and the data will be written into the database.

Edit-student.cgi

This is used to edit the student details. The administrator inputs the student number and it will bring the student details and allow any changes that need to be made.

Search-student.cgi

This script allows one of the student details to be used to search a student.

Show-student.cgi

This script accepts the student number as the input and returns the timetable of the student.

Email-student.cgi

This is used to send email to all the students on the system for important announcement of any changes.

3.2.2 Units Management

Add-unit.cgi

This is used to add the details of a unit like the unit name and unit number to the database.

Edit-unit.cgi

This script allows the administrator to edit the details of the unit.

Search-unit.cgi

This is used to search for a unit by entering any keyword for the unit name.

Total.unit.cgi

This script is used to show the total number of place for each class type in a unit.

3.2.3 Classes Management

Add-class.cgi

This script adds all the details of a class such as class id, unit index, day, time, location, type etc to the appropriate tables in the database.

Edit-class.cgi

This is used to edit the details of a class at a time.

Edit-classes.cgi

This is used to edit all the details of all classes for a particular unit. The administrator can update the details. It is mainly used for updating the quota for each class.

Show-classusage.cgi

This is used to display the class usage information for a particular unit.

Del-class.cgi

This is used to remove a particular class from a unit.

Email-class.cgi

This script is used to email the students for a particular unit or a class.

3.2.4 Waiting List management

Waitlist-unit.cgi

This shows the students on the waiting list for a particular unit.

Waitlist-edit.cgi

This is used to move the students on the waiting list for a particular unit to the new classes the administrator creates. All the students are listed on the top with a option box next to each one of them. The administrator ticks the boxes to select students to move classes. At the bottom, there is a box to send an email to those students who are moved to the new class.

Waitlist-del.cgi

This is used to remove the students from the waiting list.

Waitlist-all.cgi

This script lists all the students for every unit on the waiting list.

Waitlist-email.cgi

This is used to email all the students on the waiting list.

3.2.5 Statistics/Report

Showall-unit.cgi

It is used to list all the students (student name, student number, email address) and the total number of students for a particular unit.

Showall-students.cgi

This is used to list out all the students on the system.

Showall-overload.cgi

This is to list out all the students who are doing 5 units.

3.2.6 Download/Upload

Upload.cgi

This script allows bulk upload of class, unit and student information to the database. A common delimited format file is used as input to upload to the web server via the web interface.

Download-unit.cgi

This is used to download a list of students in a class or a unit.

Download-units.cgi

This is for downloading all the class usage information of all the units.

3.3 Tables Used in the Timetabling Database

The following tables have been used in the timetabling system database.

Tuttimes: PRACID, DAY, TIMESTART, TIMEEND, LOCATION, TYPE, ACTIVE, QUOTA, RUNWITH, NOTES

Unitpracs: UNITIDX, PRACID

Units: UNITIDX, NAME

Enrolment: STNUMBER, UNITIDX, SEMESTER

Prac enrolment: STNUMBER, PRACID, STATE

Pins: IDNO, PIN, NAME, EMAIL

Waitlist: STNUMBER, UNITIDX

4. Conclusions and Future Work

This paper first has described various interfacing mechanisms between the Web and corporate databases. Then features of various web-database interfacing mechanisms have been presented. Finally, a web-based timetabling system implemented by using one of the interfacing methods has been described. Certain changes and improvements in the coming version have been identified. These include enrollment procedure, session management, portability, user interface, security, and performance

We are interested in investigating performance implications of different web-to-database interfacing methods. This is vastly different from the general database system benchmark, general Web server benchmark, and WWW caching performance measurements reported in literature (Lazar, 1997; Saleeb, 1997). The performance measurement factors include connections per second, throughput (bytes per second), average response time (round-trip time), error rate, and web overhead ratio. Typical web-based database application usage will also be simulated. Currently, we are also working on ways to minimise or eliminate potential factors that may influence the performance measurements.

Currently, the system is being ported to Windows NT with ASP in Perl and MSQL 7.0 for the studies of implementation issues and performances comparisons. Session management and stored procedures are included.

Future work would include implementing the same application using other different interfacing methods discussed in this paper and conducting experiments to provide comprehensive quantitative performance comparisons. In addition, porting the application to different platforms would produce valuable feedback to the refinement of performance measurements.

References

Ashenfelter J. P. "Web Database Development Tools, Metaphorically Speaking," *WebNet Journal*, April-June 1999, pp. 14-15, 23.

Carriere, J., and Kazman, R. "WebQuery: searching and visualizing the Web through connectivity," *Proc. Of the 6th International WWW Conference*, 1997, pp. 701-711

Duan, N. N. "Distributed Database Access in a Corporate Environment Using Java," *Proceedings of the 5th International World Wide Web Conference*, May 6-10, 1996, Paris, France.

Feng, L., and Lu, H. "Integrating Database and Web Technologies," *International Journal of World Wide Web* (1:2), 1998, pp. 73-86.

Frey, A. "Web-to-database communication with API based connectivity software," *Network Computing* (7:18), 1996.

Hamilton, G. et al. *JDBC Database Access with Java: A Tutorial and Annotated Reference*, Addison Wesley, 1997.

Hunter, J., and Crawford W. *Java Servlet Programming*, O'Reilly, 1998.

Kim, P. C. "A Taxonomy on the Architecture of Database Gateways for the Web," *Proceedings of The 13th International Conference on Advanced Science and Technology (ICAST97)*, 1997, pp. 226-232.

Lazar, Z. P., and Holfelder, P. "Web Database Connectivity with Scripting Languages," *Web Journal* (2:2), 1997.

Murugesan, S., Deshpande, Y., Hansen, S., and Ginige, A. "Web Engineering: A New discipline for Development of Web-based Systems," *Proceedings of the first ICSE Workshop on Web Engineering, International Conference on Software Engineering*, Los Angeles, May 1999. [URL: <http://fistserv.macarthur.uws.edu.au/san/icse99-webe/>]

PHP. [Online] Available: <http://www.php.net>

Reichard, K. "Web servers for database applications", *DBMS* (9:11), 1996, pp. 226-232.

Saleeb, H. "Real-Time Database Theory and World Wide Web Caching". [URL: <http://www.eecs.harward.edu/~saleeb/projects/265.html>] (Harvard Univ., 1997)

Stein, L., and MacEachern, D, *Writing Apache Modules with Perl and C*, O'Reilly, 1999.

Wille, C. *Unlocking Active Server Pages*, New Riders, 1997.

Wong, W. "Back-end Web Databases (Making corporate data available through Web servers)," *Network VAR* (5:12), 1997, pp. 67-72.

Zhao, W. "A study of web-based application architecture and performance measurements," *Proceedings of 99' Australian Web Conference (AUSWEB99)*, 1999, pp. 50-58.