Developing a Web-Enabled Database System: An Academic Project

Paul I-Hai Lin - Indiana University-Purdue University Fort Wayne
Melissa Lin - University of Florida
Aik Mon - GE Industrial Systems

ABSTRACT

Database and Web integration offers a cost effective way to provide value added access to data stored at multiple locations through a Web browser. This paper discusses two methodologies for implementing a remote database query systems using Web browsers.

I. Introduction

In May 2000, we developed a Web-enabled alumni database for remote access by our faculty and staff. The alumni database was created from a spread sheet that contains the following fields: Title, Last Name, First Name, Middle Initial, Degree Received, Major, Graduation Date, Address, City, State, Zip code, Email address, Phone, Employer, ID, Last Gift Amount, Last Gift date, and Total Gift Amount. The developed Web-based alumni database has the following capabilities:

- View data records for all alumni (20 records per screen)
- Find a specific alumnus or group of alumni by their Last Name, First Name, Graduation Date, City, State, or Employer
- Update specific fields on alumni data records
- Update, delete, and add alumni records to database

Although the data warehouse and Web-enabled databases are well known and matured in commercial sector, for us, this is a new challenging and requires additional research. This paper is not intended to provide a comprehensive study report of tools and architectures for building database and Web integration systems. However, the Internet protocols, languages and tools, databases, Web servers, CGI, and server-side scripting are highlighted in various sections to help the selections of programming tools, system platform, Web servers, and databases for this Web-enabled alumni database system.

In this paper, we first review Web programming languages and tools needed for implementing a Web-based database in Section II. Section III gives an overview of Web-enabled database architecture. Section IV addresses Web-enabled database design. Section V presents two

methodologies for implementing the Web-enabled alumni database: Visual Basic/CGI/ODBC, and ASP/ODBC. Our presentation concludes in Section VI with a summary of findings.

II. Web Programming Languages and Tools

HTTP Protocol, HTML, and XML Languages. A Web site is a collection of HTML documents managed by a program called Web server using HTTP protocol with a default TCP port 80. HTTP has been used for global information service since 1990. A Universal Resource Locator (URL) accesses HTML documents in a specific format (Method://Host:Port/Path), for example <http://www.purdue.edu/index.html>, specifies the homepage of Purdue University using HTTP protocol with the default TCP. HTTP is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol, that can be used for many tasks beyond hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers [1].

As specified in the article "XML", the Extensible Markup Language, from World Wide Web Consortium [2], the XML is the universal format for structured documents and data on the Web. The goal of XML protocol is to develop technologies, which allow two or more peers to communicate in a distributed environment, using XML as its encapsulation language. The XML allows a layered architecture on top of an extensible and simple messaging format, which provides robustness, simplicity, reusability and interoperability.

Programming Tools for Web Server and Database Integration. Over the years, major system programming languages evolved from ALGOL, PL/1, Pascal, C, to C++. System programming languages were designed with strong-type for dealing with complex algorithms or data structures from scratch. Scripting languages, such as Perl, Python, Tool Command Language (TCL), VB scripts, Unix shells, and PHP were designed as tools for providing rapid application development and to help gluing or simplifying the connections among existing application components. The growth of the Internet has greatly expanded the popularity of scripting languages on both client-side and server-side applications. For example, Perl-CGI scripts, Java Servlet, and Microsoft Active Server Pages (ASP) allow a Web server to extend its functionality to interact with other program modules, subsystems, and databases. Figure 1 lists available framework, programming languages, and database driver types for Web-enabled database applications.

Figure 1. Server-Side Programming and Framework

Examples of server-side scripting using Perl and VBScript, ASP, and Java Servlets, Java Server Pages (JSP) can be found in [3]. CGI scripts are usually written in PERL, C/C++, Visual Basic, TCL, Python, and PHP and are usually called or invoked through a HTML form. A major disadvantage of CGI program is that the overhead of process creation for each request.

Microsoft ASP is a server-side scripting environment or framework that you can use to create and run dynamic, interactive Web server applications. With ASP, you can combine HTML pages, VBScript and other script commands, and COM components to create interactive Web pages or powerful Web- based applications that are easy to develop and modify.

Servlets are compiled Java classes for extending the functionality of a server. Servlets have full access to Java's advanced features-database connectivity, network awareness, object orientation, and built-in support for multithreaded process. There are two Java APIs for servlets: *javax.servlet.**, and *javax.servlet.http.**. Servlets are appropriate for performing the following tasks: manipulating HTTP status codes and headers, using cookies, tracking sessions, saving request information, accessing database, business logic, etc.

Java Server Pages (JSP) use special HTML tag inside Web pages for generating dynamic content. It helps to separate applications from the appearance of the Web page and to increase the speed of the development process. A JSP file is named with a .jsp extension and may include and work with the following program modules and reusable components: Java code, applets, servlet, JavaBeans, and Enterprise JavaBeans.

III. Web-Enabled Database Architectures

With the popularity of the Internet and intranets, an access to a large number of data that stored in Database Management Systems (DBMS) through the Web for electronics publishing, electronic commerce, and enterprise applications is increased. The four-layer Internet architecture consisting of presentation, content, application, data and service layers is considered [4].

Figure 2 depicts a well known three-tier Web-enabled database architecture [5]. It consists of a client tier, a middle tier, and a data tier. Relational database systems (RDBMs) evolved in the 1970s and flourished in the 1980s. Structured Query Language (SQL) is the universal language for client-server execution and remote database access. SQL provides both data definition language (DDL) and data manipulation language (DML). Examples of popular RDBMs include Oracle, IBM DB2, Informax, Microsoft SQL, Sybase, etc.

Figure 2. Three-Tier Web-Enabled Database Architecture

Open Database Connectivity (ODBC), created by Microsoft [6], is a set of standard APIs that allow a programmer to embed SQL statements for accessing data from a variety of relational database systems. It is based on the Call-Level Interface (CLI) specifications from X/Open and ISO/IEC for database APIs and uses Structured Query Language (SQL) as its database access language.

Java Database Connectivity (JDBC) technology defines representations of SQL interfaces for Java programs to access virtually any tabular data source across a wide range of SQL databases.

IV. Web-Enabled Databases Design

The objectives of the desired Web-based database system should include all services needed in such applications:

- Query databases
- Generate dynamic HTML documents
- Databases updating

Figure 3 illustrates basic steps in querying Web-enabled databases using either ODBC, or JDBC technology.

Figure 3. Basic Steps in Querying Web-Enabled Databases

Web-Enabled Database Staffing Requirements. It is necessary to review the programming skill and knowledge related to the project of all team members before the project. We found that all team members also have had experiences with HTML language, JavaScript, VBScript, Perl, C/C++, and Visual Basic for developing Web-based applications. All members had Microsoft Access database experience. One team member has extensive IBM DB2 experience and is familiar with Erwin database design and modeling tools.

Database and Web Server Selection. There was also a great deal of uncertainty as to whether the Omni Web server running on a Windows 98 PC, that also served the lab activities of the Web Programming Application course could satisfy this need. Windows 2000 was released but was not a mature platform at this time.

Since there is only one table in the alumni database with the following fields: Title, Last Name, First Name, Middle Initial, Degree Received, Major, Graduation Date, Address, City, State, Zip

Code, Email Address, Phone, Employer, ID Field, Last Gift Amount, Last Gift Date, and Total Gift Amount. The original data table in Microsoft Excel file format of about 260 Kbytes can be easily converted to Microsoft Access 2000 database. We estimated that only a small number of concurrent users (less than 10) are expected. In selecting a database, the design team decided that the Access 2000 would be sufficient to handle this task. Many public Web sites require SQL servers since they would allow many users who are simultaneously updating, adding, and deleting records. Microsoft SQL would be reserved for future upsizing.

The design team also decided that the PC running a Windows 98 operating system and Omni Web server would be sufficient to work as a gateway to the alumni database. Server-side database would be a good choice to minimize the cost. There is no need at this time to design a separate database server since it can be easily upgraded to Windows 2000, Microsoft Internet Information System or Apache, with Microsoft SQL.

Selecting Server-Side Programming Tools. The project team selected both Visual Basic and ASP as server-side programming tools based on the following factors:

- Visual Basic supports Microsoft Access database files as its native format
- With ODBC support, Visual Basic can access many other database formats such as SQL server, Oracle, and DB2
- Visual Basic comes with Visual Data Manager for database design
- ASP uses VBScript which is a subset of Visual Basic

Microsoft Visual InterDev was selected as the main development tool for this project due to its capability of supporting database development, Web page design, and Web site management.

V. Implementation

Initially, the Web-enabled database application was written in Visual Basic due to the team member's experience with this language. It would required more time for us to implement the same Web-database system using the ASP. After the VB-version was tested successfully, the second version of the application was written in ASP. Both methodologies require that all team members have a working knowledge of CGI (Common Gateway Interface) protocol for use in CGI scripts.

Visual Basic Version

Data Flow of Web-enabled Database System Using Visual Basic. Figure 4 shows the data flow diagram, which includes one static page with forms for login to the system, and two Visual Basic-based CGI programs for performing authentication, and database accessing activities. This diagram also serves as the Web delivery design diagram.

Figure 4. Data Flow Diagram of the Web-enabled Alumni Database System

Login Page. The source code of the login page as shown in Figure 5 was designed to include some JavaScript codes for displaying time and date information. A fragment of the EETAlumni.html is shown below:

```

<HTML>
<HEAD><TITLE>Verifying a username and a password</TITLE>
</HEAD>
<script LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
dayName = new Array; ("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday");
monName = new Array ("January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December");
now = new Date;
</script>
<BODY bgcolor = "#ffffff" onload="document.forms[0].username.focus()">
...
<FORM action="http://149.164.36.204/cgi-bin/EET_Alumni1/PassWord.exe" method="post">
<TABLE cellpadding="1" width="170" border="0">
  <TR>
    <TD><INPUT name="action" type="radio" Value="unchangepw" Checked><STRONG><FONT
      color=#333399>UnChange Password<br>
    <INPUT name="action" type="radio" Value="changepw"></FONT></STRONG><FONT
      color=#333399><STRONG>Change Password<br></STRONG></FONT>
    <TR><TD><STRONG><FONT color=#333399><FONT size="2" face="Arial">UserName:</FONT>
      </FONT></STRONG></TD>
    <TD><INPUT name="username" size="16" maxlength="16" ></TD></TR>
    <TR><TD><STRONG><FONT color=#333399><FONT size="2" face="Arial">PassWord:</FONT>
      </FONT></STRONG></TD>
    <TD><INPUT type="password" name="password" size="16" maxlength="16"></TD></TR>
    <TR><TD>&nbsp;</TD><TD><INPUT type="submit" value="Login"></TD></TR></TABLE>
  
```

```

</FORM> </DIV>
<center>
<p><script LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
document.write("<H3>Today is " + dayName[now.getDay()] + ", " + monName[now.getMonth()] + " " +
now.getDate() + ".");
. . .
<FONT
color=#333399>Problems</FONT>:<A href="mailto:eet499@hotmail.com"> Send e-mail</A>
<hr width="100%" size=4 noShade color=#330099>
</center>
</BODY></HTML>

```

Figure 5. Log-in Page for Accessing the Web-enabled Alumni Database

Designing CGI Programs for Interfacing to Users and Alumni Database. The VB script PassWord.exe will be called through the form to verify the user. It also allows the user to perform the tasks as shown in the Figure 6 which is generated dynamically by this PassWord.exe. The user enters all the desired information into this page, then clicks the send button to request service from EET_Alumni.exe CGI program. This CGI program is assuming the tasks of retrieve form data from CGI variables, connect database, form query statement, make a query to retrieve data records, and finally format data record in HTML page format for display. A fragment of the EET_Alumni.bas source code is shown below:

```

Dim db As Database           'define database
Dim rs As Recordset         'Recordset

'define variable that will convert from Html variable
SendHeader "IPFW, ECET Alumni List Database"
'Open Database

```



```

Set db = OpenDatabase("C:\EET_Alumni1\EET_Alumni1.mdb")
Set rs = db.OpenRecordset("EET_Alumni1")

'Check Update file, this statement will execute when updating individual file
If sUpdate = "update" Then
    UpdateMember
End If
Select Case sAction 'retrive the Selection from the user seclection option
Case "find"
    Call FindMember      'Call the display function for table
Case "addmember"
    Call AddMember
Case "deletemember"
    Call DeleteMember
Case "findindi"
    Call FindIndividual
End Select

SendFooter      'Send Footer at the end of program
rs.Close        'Close the recordset
db.Close        'close the database program

```

The image shows two side-by-side browser windows from Microsoft Internet Explorer. The left window, titled 'ECET Alumni List - Microsoft Internet Explorer', displays the main interface of the 'Electrical and Computer Engineering Technology Alumni List'. It includes sections for 'Select Main Options' (with radio buttons for 'View All Members' and 'Search By: LastName'), 'Select Display Objects Options' (with radio buttons for 'Table' and 'Form'), and 'Select Display Fields' (with radio buttons for 'All Fields' and specific field selections). The right window, also titled 'ECET Alumni List - Microsoft Internet Explorer', shows the 'Select Options to write' form. This form contains several input fields for member information, including Member ID#, Title, Last Name, First Name, Middle Name, Degree, Major, Graduate Date, Address, City, State, Zip, Phone, Email, Employer, Last Gift, and Total Amount. The form also includes 'Send' and 'Reset' buttons and a 'Help' link.

Figure 6. The Dynamic Page Generated by the VB Script: PassWord.exe

Active Server Page Version

ASP Programs for Interacting to Users and Database Interfacing. Three asp files named AlumniVerifying.asp, MainMenu.asp, and AlumniList.asp were written for this Web-enabled alumni database application as well. The dataflow flow diagram as shown in Figure 7 depicts how those three asp programs interact to a user.

Figure7. ASP Version Dataflow Diagram of the Web-enabled Database

The AlumniVerifying.asp and MainMenu.asp programs are written and are available for access through the TCP port 81 of our Web server. As shown in Figure 7, AlumniVerifying.asp provides a gateway for users to login to the system and the MainMenu.asp allows the user to input query and invokes the AlumniList.asp program to access the alumni database.

Figure 8. Two Screens Showing the Execution of AlumniVerifying.asp and MainMenu.asp programs

Soon after a user is authorized to access the system, the MainMenu.asp program then sends out a dynamic page, as shown in right-side screen of Figure 8, for the user to perform one of the following tasks:

1. View all members of the alumni database, and display 20 records at a time
2. Search for a member or a group of members by their Last Name, First Name, Graduation date, Employer, City, or State.
3. Use Help.html page to learn more about how to enter the search criteria
4. Enter new member information or update existing member information

Code fragments of the MainMenu.asp and AlumniList.asp pages are listed below for reference:

```
<!-- MainMenu.asp -->
<HTML><HEAD><TITLE>ECET Alumni List</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--
-->
</SCRIPT>
<% @Language = VbScript %>
<%
Dim Connection, Rs, sSql
Dim sPassWord, sUserName
sUserName = Request.Form("username")
sPassword = Request.form("password")
sSql = "Select * From tblUserInfo WHERE UserName = '" & sUserName & "' And Password = '" &
sPassword & "'"
Set Connection = Server.CreateObject("ADODB.Connection")

Connection.Provider="Microsoft.Jet.OLEDB.4.0"
Connection.Open "C:\Inetpub\wwwroot\WebAutomation\EetAlumni.mdb"
Set Rs = Server.CreateObject("ADODB.Recordset")
Rs.Open sSql,Connection
'Connect to the database
</HEAD>
<BODY>
<% 'If success then
IF Not RS.Eof Then
Response.Write "<FONT color=darkblue size=5><STRONG>Electrical and Computer Engineering" & _
"Technology Alumni List</STRONG></FONT><br>" & _
"<hr width=100% size=6 noShade color=#330099>" & _
...
...
<HTML>
```

```
<!-- AlumniList.asp -->
<% @Language = VbScript %>
<%
Dim Connection, Rs, sSql, bFlag
Dim sTitle, sLastName, sFirstName, sMiddleName
Dim sDegree, sMajor, sGradDate, sAddress, sCity
Dim sState, sZip, sEmail, sPhone, sEmployer, sID
Dim sLastGiftAmt, sLastGiftDate, sTotalGiftAmt, sAction
Dim sMain, sSelect, sSearchText, sDisplayStyle, sDisplayFiled
sMain = Request.Form("Main")
sSelect = Request.Form("Select")
sSearchText = Request.Form("SearchText")
sDisplayStyle = Request.Form("object")
sDisplayFiled = Request.Form("field")
sTitle = Request.Form("Title")
sLastName= Request.form("LastName")
sFirstName = Request.Form("FirstName")
sMiddleName= Request.form("MiddleName")
```

```

...
...
sID = Request.Form("ID")
sLastGiftAmt= Request.form("LastGiftAmt")
sLastGiftDate = Request.Form("LastGiftDate")
sTotalGiftAmt= Request.form("TotalGiftAmt")
sAction = Request.Form("Action")
Set Connection = Server.CreateObject("ADODB.Connection")
Connection.Provider="Microsoft.Jet.OLEDB.4.0"
Connection.Open "C:\Inetpub\wwwroot\WebAutomation\EetAlumni.mdb"
Set Rs = Server.CreateObject("ADODB.Recordset")
Select Case sMain
Case "allView"
    Call ViewAllMember
Case "searchby"
    Call ViewSearchBy
case else
    Call ViewAllMember
End Select
%>
</HEAD><BODY>
<%
Sub ViewAllMember()
sSql = "Select * From tblEetAlumni"
Rs.Open sSql, Connection
%>
...
...
</BODY>
<%
Set rs = Nothing
Set cmd = Nothing
Connection.Close
set Connection = nothing
%>
</HTML>

```

VI. Summary

This paper presents an overview of HTTP protocol, HTML/XML languages, Web server platforms and server-side programming tools, and Web-enabled database architecture. We also address important issues concerning Web-enabled database design and deployment - staffing requirements, database and Web server selection, server-side programming tool selection, Web client design, and CGI program design. We have had the Web-enabled alumni database in use since July 2000 and we have upgraded Windows 98 to Windows 2000. Authorized users are satisfied with this Web-enabled alumni database system. The experience learned from this project helped all team members in various ways. The student member was able to secure his current job a system engineer in GE industrial system. The staff team member is now able to apply the knowledge gained from the project to develop JDBC-based Web-enabled database system applications for University of Florida. And, the faculty team member is able to offer more advanced courses that deal with enterprise applications and industrial automation applications at Purdue University Fort Wayne Campus. Currently, we are redesigning this system to include Windows 2000 advanced server with Internet Information Server, SQL 2000 database. The Web server and database programming environments will be ASP.NET and ADO.NET.

References:

- [1] Hypertext Transfer Protocol -- HTTP/1.1, <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [2] World Wide Web Consortium, www.w3c.org
- [3] Deitel, Deitel, and Nieto, Internet & World Wide Web How To Program, Prentice Hall, 2nd edition, 2001.
- [4] Miroslav Benda, "Internet Architecture: Its Evolution from an Industry Perspective," March/April 1998, IEEE Internet Computing, pp. 32-35.
- [5] Raghu Ramakrishnan and Johannes Gehrke, "Chapter 22 Internet Databases," Database Management Systems, 2nd edition, McGraw Hill.
- [6] Microsoft ODBC 3.51, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/htm/dasdkodbcoverview.asp>.
- [7] The Open Group, Distributed Computing Environment Overview, <http://www.opengroup.org/dce/info/papers/tog-dce-pd-1296.htm>.
- [8] Scott Seely, SOAP Cross Platform Web Service Development Using XML, Prentice Hall, 2002.
- [9] Simple Object Access Protocol 1.1, <http://www.w3.org/TR/SOAP/>
- [10] The Sun Java Soft, JDBC Database Access API, <http://java.sun.com/products/jdbc/index.html>.
- [11] XML Protocol Activities, World Wide Web Consortium, <http://www.w3c.org/2000/xp/>.
- [12] Web Service Description Language (WSDL) 1.1, W3C Note 15, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.
- [13] The UDDI (Universal Description, Discovery, and Integration) Technical White Paper, September 6, 2000. http://uddi.microsoft.com/developer/tech_white_paper.doc.
- [14] Una Kearns, "eContent Services - The Service Behind Web Services," Web Services Journal, November 2001, Volume 1, Issue 2, pp. 7-12.
- [15] Phil Karecki, "Web Services Fundamentals: From Whence It Came," Web Services Journal, November 2001, Volume 1, Issue 2, pp. 18-20.
- [16] Rebecca Diosrecki, "Web Services: Enabling Technology or New Programming Paradigm," Web Services Journal, November 2001, Volume 1, Issue 2, pp. 22-24.
- [17] Performance Assurance: An Architectural-Approach to Meeting Server-Side Java Performance Requirements, Java Report, April 2001, pp 36-43.
- [18] Cynthia Mcfall, "An Object Infrastructure for Internet Middleware, IEEE Internet Computing," March/April 1998, pp. 46-51.
- [19] Jim Conallen, Building Web Applications with UML, 2000, Addison Wesley.

Biographies

PAUL I-HAI LIN

Paul is Associate Professor and Chair of the Electrical and Computer Engineering Technology, at Indiana University - Purdue University Fort Wayne. He has been with Purdue University since 1985 and is a registered Professional Engineer (EE) in California and Indiana. Prior to joining IPFW, he taught in the Engineering and Technology Department of Dutchess Community College (NY) for three years; in the Electrical Engineering Department of National Taipei Institute of Technology for two years; and worked in industry for 8 years. He was a Visiting Associate Professor in the Electrical Engineering Department of National Taipei Institute of Technology, from January-July 1995, and the Conference Vice-Chair of the 1995 IEEE International Conference on Industrial

Automation and Control at Taipei. He is a senior member of IEEE and was the Chairman of the Manufacturing System Development and Application Department of IEEE-Industry Applications Society from January 1998 to December 1999. Lin's current interests include Web engineering, distributed intelligent control of embedded real-time systems, and sensors in industrial control applications.

MELISSA LIN

She is a database administrator in the Office of Academic Affairs – Information Technology at the University of Florida. She is currently in charge of implementing the data warehouse for the University of Florida. She has designed the first data warehouse for the University of Florida last year. Previously, she was a Visiting Professor in the Department of Computer Science at Indiana University - Purdue University Fort Wayne during 1999. Before that, she worked as systems analyst at GTE Data Service, Fort Wayne, Indiana, for 11 years and was responsible for E911 and CBSS Billing support. Lin's current interests include modeling for data warehouse and Web-based user interface applications.

AIK MON

Aik Mon received his B.S. in Electrical Engineering Technology from Purdue University, Fort Wayne Campus. He is a Systems Engineer at GE Industrial Systems, Fort Wayne. His main responsibility is to maintain and support TIES (Transformer Integrated Engineering System) for designing new products, providing bill of materials, and estimating the cost of the product. He also serves as a database administrator, a Web developer, a Visual Basic developer, and a database front-end and back-end developer.