UC Irvine
UC Irvine Previously Published Works

Title

Accessing existing business data from the World Wide Web

Permalink
https://escholarship.org/uc/item/1b49s075

Journal
IBM Systems Journal, 37(1)

ISSN
0018-8670

Authors
Peng, C.-S.
Chen, S.-K.
Chung, J.-Y.
et al.

Publication Date
1998

DOI
10.1147/sj.371.0115

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, availalbe at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Accessing existing business data from the World Wide Web

by C.-S. Peng
S.-K. Chen
J.-Y. Chung
A. Roy-Chowdhury
V. Srinivasan

The World Wide Web has experienced phenomenal growth over the last few years. Although, at its inception, Web technology was primarily used to retrieve information stored in static documents, important current uses of the Web include retrieval of dynamically changing information and the conducting of business transactions. Such uses of the Web result in access to dynamically changing data on or through Web servers, usually stored in a database. Huge volumes of business data exist on mainframes and other mature platforms that cannot be moved to client/server or workstation-based platforms, due to cost or performance issues. Providing Web access to these legacy data, therefore, is of great commercial interest to businesses. In this paper, we survey several solutions that have been developed to access existing business data through the Web. We discuss the details of two solutions developed at IBM: DB2® (DATABASE 2™) World Wide Web Connection and Net.Data™. Each of these is a pure middleware approach as opposed to approaches that are integrated with either the Web server or the database management system, which accounts for their flexibility and power.

Since the development of the first graphical World Wide Web (or Web) browser for the Internet, Web applications have become the most popular method of information retrieval from the Internet. The main advantages that Web application technology offers over other methods of information retrieval from the Internet are its ease of use (via a simple mouse point-and-click interface), the ability to retrieve different kinds of data and documents (including multimedia documents), and the ability to organize related information via hyperlinks.

The first generation of Web protocols and applications only allowed the retrieval of static hypertext files from a Web server. However, with the introduction of the Common Gateway Interface (CGI)[1] specification for dynamic execution of programs on a Web server, the potential usefulness of the Web was greatly increased. End users now can interact with dynamic Web applications to access information in databases or conduct searches, for example. The most recent enhancement to the power of the World Wide Web has been the introduction of Java**,[2] a programming language that can be used to develop completely portable Web applications. Programs called "applets" written in Java can be downloaded by a Web browser after connecting to a Web server and can be used to extend browser or server functionality in powerful ways.

Another use for Web technology is from the emerging intranet (corporate internet or internal internet). The intranet allows a company to make its applications and information systems immediately available

©**Copyright** 1998 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8670/98/$5.00 © 1998 IBM

to all its offices on heterogenous client platforms as long as they support Web browser technology. The introduction of the "thin client" or "network computer"[3] will likely hasten the move toward corporate intranets where powerful Web servers will be used to distribute application software to thin clients as needed, which will result in lower costs due to group collaboration, greater application sharing, and less expensive clients.

Database management systems (DBMSs) and on-line transaction processing (OLTP) applications are at the heart of business processing and have been deployed by enterprises for decades. Examples are IBM's DB2* (DATABASE 2*) DBMS,[4] CICS*[5] (Customer Information Control System), and IMS* (Information Management System).[6] These commercial systems meet the customer's requirements for reliability, scalability, security, interoperability, and performance. With Web applications having become the preferred mode of on-line information retrieval, it is extremely important from a business perspective to integrate DBMS and OLTP with the Web technology, thereby allowing clients to conduct business through Web applications. From the point of view of the business organization, enabling access to existing data by Web applications is the way to take advantage of the ubiquitous Web environment without massively re-engineering the database and application codes. The benefit for users is to get consistent, reliable service through a much more friendly interface.

In the following sections, we will show how to put the pieces together to achieve the goal of accessing existing business data from the World Wide Web. The paper is organized as follows. In the next section we give a brief introduction to the protocols used on the World Wide Web and an overview of major approaches to the Web enablement of existing business data. Next we present IBM's approaches to accessing existing business data from the Web browser by means of different gateways. Two Web gateways for accessing DB2 data, DB2 World Wide Web (DB2 WWW) Connection and Net.Data*, are presented. Finally we report on some related work in the industry and end with some conclusions.

Web enablement for enterprise computing

Since the Web client/server architecture plays an important role in the Web enablement for enterprise computing, we first review the fundamentals of the Web technology, then present the approaches to integrate it with DBMSs.

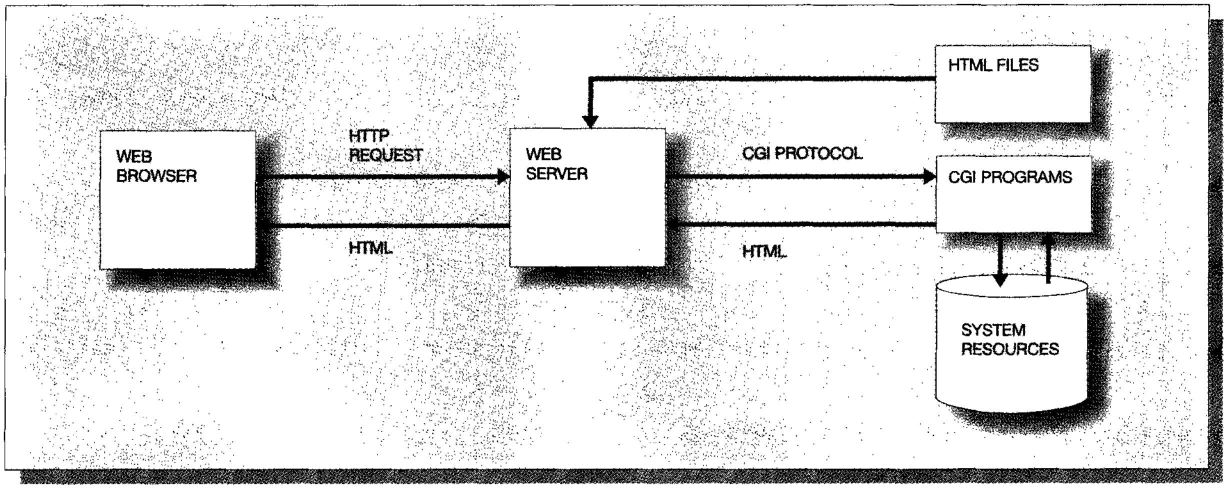
Web fundamentals. The World Wide Web is supported primarily by three main standards: uniform resource locator (URL), HyperText Transfer Protocol (HTTP),[7] and HyperText Markup Language (HTML).[8,9] These standards are used by all Web browsers and servers to provide a consistent mechanism for locating, retrieving, and displaying information. Although none of the above-mentioned standards specifies or relies on a specific transport mechanism, Transmission Control Protocol/Internet Protocol (TCP/IP) is the transport mechanism used for communications between Web clients and servers, largely due to its being the protocol of choice for communication over the Internet. More recently, a standard called Common Gateway Interface (CGI)[1] has been developed for interfacing dynamic applications to Web servers. In the remainder of this section, we briefly describe each of these standards.

A URL is a simple addressing scheme that uniquely identifies a document or file regardless of the protocol. It contains the accessing protocol name, the server and the port to connect to, and the file path. It may include the other specific information such as an application or shell script name, and possibly the input arguments. The URL can be used to specify not only the HTTP protocol but also a common set of well-established protocols for the Internet such as File Transfer Protocol (FTP), Gopher, Telnet, Simple Mail Transfer Protocol (SMTP), Network News Transfer Protocol (NNTP), Wide Area Information Service (WAIS), etc. All of the services are built on top of the client/server communication model, between browsers and servers. The Web browser breaks down each URL link request into its constituent components and uses the protocol section to determine how to proceed.

HTTP[7] defines interactions between the client browser and the HTTP server. It is a flexible, application-level communication protocol that is used to transfer data of various types (as specified by the Multipurpose Internet Mail Extensions, or MIME format) from Web server to browser. It is a stateless protocol in which each connection is established for the purposes of retrieving exactly one object, for example an HTML page or a Graphics Interchange Format (GIF) file.

HTML[8] is derived from the Standard Generalized Markup Language (SGML) that defines areas of textual information by tagging them with specific formatting information. Tags are defined functionally rather than visually. Each browser can interpret the

Figure 1 Common Gateway Interface (CGI) interaction



tags according to its configuration settings, supported fonts, and windowing environment. HTML also provides the ability to create hypertext links between documents and within a document.

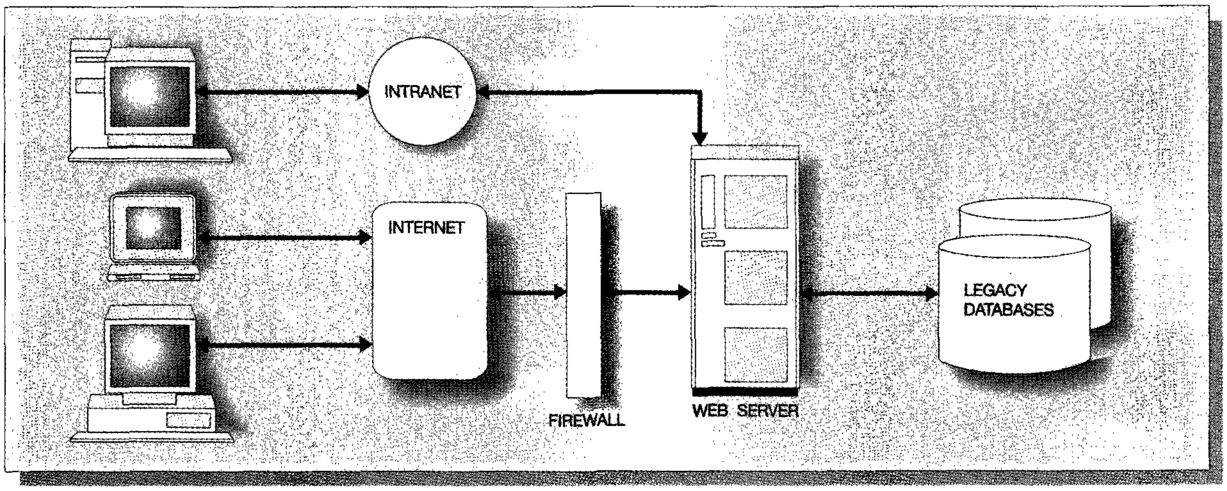
The CGI standard enables applications to return data to browsers via the Web server. Web servers implementing the CGI protocol typically treat files in certain directories on the server machine as executable programs rather than documents. Such directories are specified in the configuration file of the Web server. When a client sends a request to the Web server, the parsing of the URL by the Web server enables it to determine if the file to be retrieved is to be treated as an executable program. The URL can contain extra information to be passed to the CGI application. The Web server puts information about the server, browser, and the URL into a predefined set of environment variables and invokes the CGI application. For certain types of requests, the Web server can pass information to the CGI application using standard input or argument lists. The CGI application can then retrieve this information and customize its processing based on the information. After it is done processing, the CGI application writes a full or partial header, followed by other output, to its standard output area. The Web server completes the header, if it needs to, and passes the data written by the CGI application back to the browser. The CGI mechanism is illustrated in Figure 1.

CGI programs are external to the server and they can be implemented in a variety of languages, as long as they conform to the CGI specifications. C, C++, Java, Perl, Tcl/Tk, JavaScript**, C shell, Bourne Shell, 4GL, etc., are some of the popular languages that have been used to write CGI applications.

Web enablement for legacy data. The basic idea in placing existing business data on the Internet using Web technology is to utilize the Web server and applications invoked by the Web server as the middle tier in a three-tier client/server architecture (see Figure 2). The middle tier is then responsible for accepting the requests for database access initiated at Web browsers, initiating proper database transactions on the DBMS, collecting the results from the transactions, and returning them to users through the Web server. The firewall (as shown in the figure) serves as the security protection to prevent unauthorized access from the Internet.

The first approach is to modify the Web server directly. All the major Web server vendors have developed servers that directly support DBMSs on certain platforms. Examples are Microsoft Corporation's[10] Internet Information Server (IIS**) and Netscape Communications Corporation's[11] LiveWire Pro** that support Microsoft's SQL Server** and Informix Software Inc.'s[12] Online SQL database, respectively. Proprietary application programming interfaces (APIs) that can be used to perform specific database operations have also been developed for popular Web servers. Examples of such interfaces are ISAPI** for Microsoft and NSAPI** for Netscape.

Figure 2 Three-tier architecture



The second approach that major DBMS vendors are trying to provide involves Web connectivity enhancement for their own DBMS. DBMS programmers can then easily build CGI applications to access the database and handle HTML-based management by using the enhancements provided to the native language of the DBMS. R:WEB** from Microrim, Inc. and Illustra's Web DataBlade** (acquired by Informix) are examples of this approach. Microsoft's ActiveX** and Java languages from Sun Microsystems have also been recently enhanced to support database access.

The third approach is pure middleware. Vendors and third-party companies implement the middleware to seamlessly and transparently deliver the data and messages between the Web server and applications or DBMS without modifying either. Information Builders' Enterprise Data Access (EDA**) and IBM's DB2 World Wide Web Connection and Net.Data are examples of such products.

Detailed descriptions of IBM's middleware solutions will be presented in the next section, followed by brief descriptions of competing products from other vendors.

IBM's solutions for Web access to legacy data

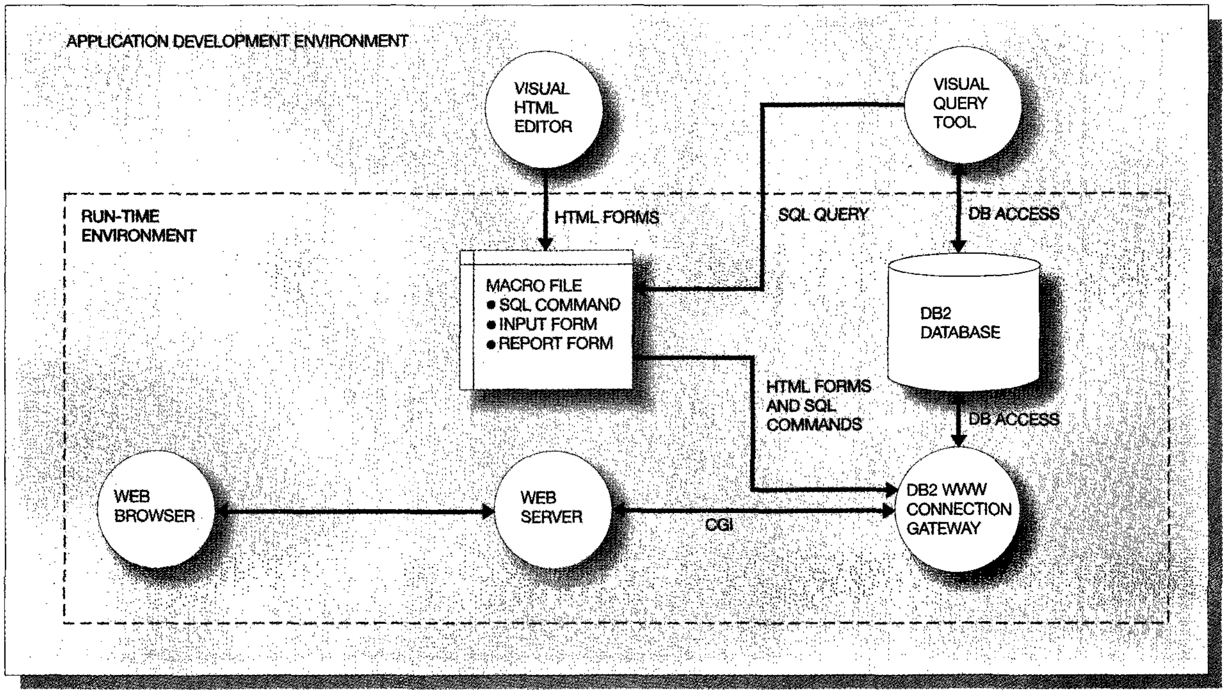
In this section, we introduce some of IBM's Web-to-database gateway implementations. We first discuss DB2 World Wide Web (DB2 WWW) Connection,[13] which enables quick and easy connection of applications that access relational DBMS (DB2) data stored on the Web. We next discuss Net.Data,[14] a successor to DB2 WWW Connection that provides greater functionality and performance than its predecessor.

DB2 WWW Connection. DB2 WWW Connection[13] is a middleware solution to the problem of accessing existing business data, in this case DB2 data, from the World Wide Web on mainframes running Multiple Virtual Storage (MVS*). It has been implemented as a single CGI application for accessing DB2. Since it is a CGI application, DB2 WWW Connection can be invoked in the usual manner by specifying the URL appropriately.

Key to application development using DB2 WWW Connection is the notion of a *macro file*. Macro files are written by application developers and are the mechanism for conveying information to DB2 WWW Connection about gathering user input, queries to be executed, and the manner in which results are to be formatted. Macro files primarily consist of three sections: HTML_INPUT, SQL, and HTML_REPORT.

A flexible variable substitution mechanism has been designed to provide cross-language variable substitution between HTML input and SQL query strings, as well as between SQL result rows and HTML output. Variables can be defined either through the URL or in the macro file, and can be used in any of the sections in the macro file.

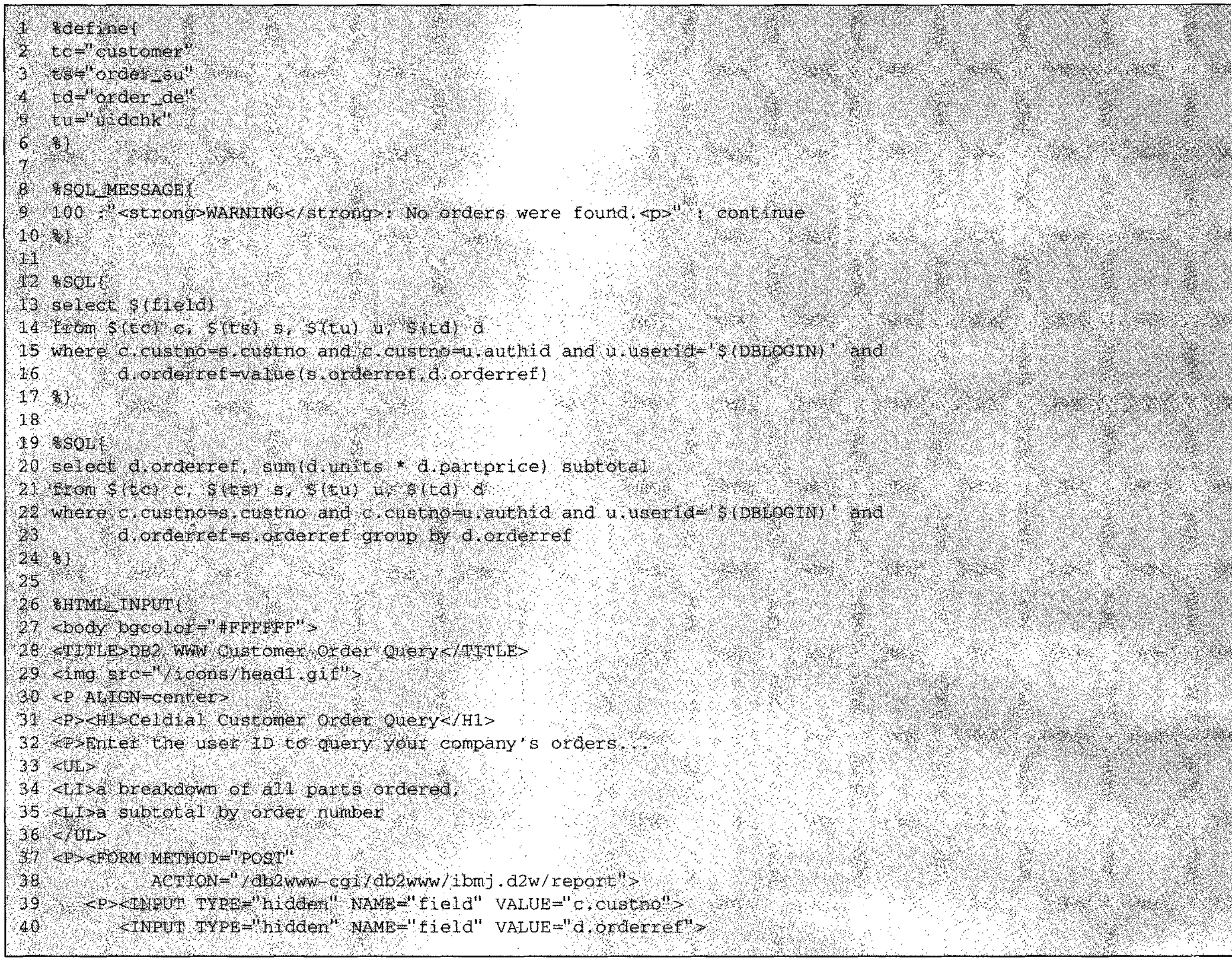
Figure 3 DB2 WWW Connection system overview



Since most of the contents of a macro file consist of either standard HTML for formatting purposes, or standard SQL for describing the kind of query to be executed against the database, it is possible to use graphical user interface-, or GUI-based tools, for creating large portions of the macro file. Also, Lotus Approach** has recently been extended to directly allow creation of macro files using a GUI front end. While some of these tools are not supported on some popular platforms for business processing, for example MVS, we can use the tools to create macro files on other platforms and then move them over to MVS. Figure 3 shows both the application and run-time environments for DB2 WWW Connection. Figure 4 is an example of a macro file. It accepts the user's selections, queries the databases, and reports the search results. There are two SQL sections (starting at lines 12 and 19). The HTML_INPUT section and the HTML_REPORT section start at line 26 and line 69, respectively. Note that the line number label is only for the purpose of demonstration.

In order to more clearly describe the functioning of DB2 WWW Connection, we go through the flow of a user's interaction with DB2 WWW Connection at run time. Figure 5 illustrates the interactions. The user invokes DB2 WWW Connection by specifying a URL. The URL also carries information about the macro file to be processed and the mode in which it is to be processed. The URL is tied to a hyperlink in an HTML page displayed by the Web browser. When the user clicks (a mouse, for example) on the link, the browser uses the information in the URL to make an HTTP request to the Web server. The Web server invokes DB2 WWW Connection in response to the request and passes it information about the macro file and the mode in which it is to be processed. Typically, the mode specified is "input," which results in the HTML_INPUT section being processed in the macro file. Usually, this results in a form being extracted out of the HTML_INPUT section and being written to standard output. The form is then picked up by the Web server and sent back to the Web browser, which displays it to the user. Typically, the action associated with the form (specified as part of the form) is for DB2 WWW Connection to be invoked on the same macro file in "report" mode. When the user fills out the form and submits it, the Web browser checks the action specified in the form and sends an HTTP request back to the Web server. The

Figure 4 An example of a macro file (DB2 WWW Connection)



```
%define{
tc="customer"
ts="order_su"
td="order_de"
tu="uidchk"
%}

%SQL_MESSAGE{
100 :"<strong>WARNING</strong>: No orders were found.<p>" : continue
%}

%SQL{
select $(field)
from $(tc) c, $(ts) s, $(tu) u, $(td) d
where c.custno=s.custno and c.custno=u.authid and u.userid='$(DBLOGIN)' and
      d.orderref=value(s.orderref,d.orderref)
%}

%SQL{
select d.orderref, sum(d.units * d.partprice) subtotal
from $(tc) c, $(ts) s, $(tu) u, $(td) d
where c.custno=s.custno and c.custno=u.authid and u.userid='$(DBLOGIN)' and
      d.orderref=s.orderref group by d.orderref
%}

%HTML_INPUT{
<body bgcolor="#FFFFFF">
<TITLE>DB2 WWW Customer Order Query</TITLE>
<img src="/icons/head1.gif">
<P ALIGN=center>
<P><H1>Celdial Customer Order Query</H1>
<P>Enter the user ID to query your company's orders...
<UL>
<LI>a breakdown of all parts ordered,
<LI>a subtotal by order number
</UL>
<P><FORM METHOD="POST"
        ACTION="/db2www-cgi/db2www/ibmj.d2w/report">
   <P><INPUT TYPE="hidden" NAME="field" VALUE="c.custno">
      <INPUT TYPE="hidden" NAME="field" VALUE="d.orderref">
```

user's choices and entries in the form are encoded as part of the URL, as are the name of the macro file and the mode in which it is to be processed (recall that this is usually "report" at this stage). The Web server invokes DB2 WWW Connection and passes it the extra information in the URL (name of the macro file, mode, and user input). The user input is used to specify variable bindings and overrides any definitions of the same variable in the macro file. When a macro file is processed in report mode, it results in the HTML_REPORT section being parsed by DB2 WWW Connection. Usually, in the HTML_REPORT section, there are references to one or more SQL sections. These are processed by the DB2 pluggable components or backend, which is a part of DB2 WWW Connection, and may result in a table being retrieved. A page consisting of the HTML statements in the HTML_REPORT and SQL sections as well as the formatted output of executing the SQL statements is written to standard output by DB2 WWW Connection,

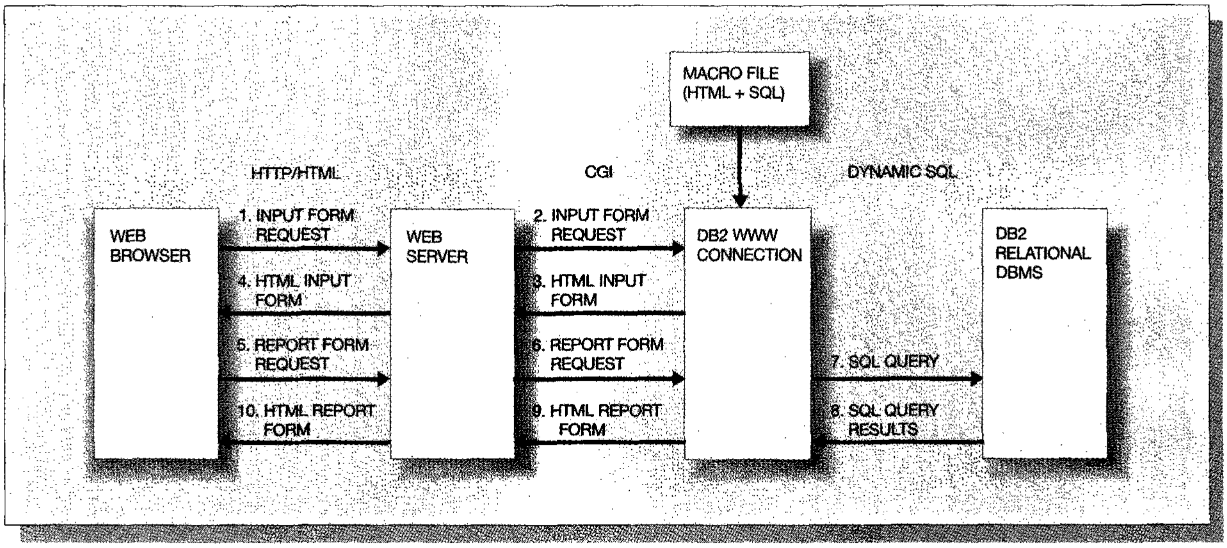
```
       Select information needed from the list, enter your user ID, and ...
    <H4> What information do you want? </H4>
    <SELECT NAME="field" MULTIPLE SIZE=4>
      <OPTION VALUE="d.partno">Part number
      <OPTION VALUE="d.model">Model number
      <OPTION VALUE="d.units">Number of units ordered
      <OPTION VALUE="d.partprice">Part Price
      <OPTION VALUE="s.orderdate">Order date
      <OPTION VALUE="s.planshipdate">Planned ship date
      <OPTION VALUE="s.status">Order status
    </SELECT>
    <P>You <strong>must</strong> input a valid user ID, ...
    <HR><PRE>
    USERID : <INPUT TYPE="text" NAME="DBLOGIN" SIZE=30 VALUE="supersoft">
    </PRE><HR>
    Show SQL statement on output?
    <INPUT TYPE="radio" NAME="SHOWSQL" VALUE="YES"> Yes
    <INPUT TYPE="radio" NAME="SHOWSQL" VALUE="" CHECKED> No
    <P><INPUT TYPE="submit" VALUE="Submit query">
       <INPUT TYPE="reset" VALUE="Reset">
</FORM>
<P>
<HR>
<B>[
     <a href="http://www.ibm.com/">IBM home page</a> | ...
]</B>
%}

%HTML_REPORT{
<body bgcolor="#FFFFFF">
<TITLE>DB2 WWW Customer Order Results</TITLE><img src="/icons/d2whead1.gif">
<BR>
<P>The following two lists are your parts...
<BR>
%EXEC_SQL
<P>If this is incorrect please call (914) 555-1234.
<HR>
<P>Other pages of interest: ...
%}
```

and is sent back to the Web browser by the Web server, which then displays the results to the user. Note that the process need not stop here, since it is possible to provide links to process other macro files in the results generated. Figure 5 shows the run-time interactions between the end user and DB2 WWW Connection. Figures 6 and 7 show the browser output resulting from processing the macro file of Figure 4 in input and report mode, respectively. More details about the DB2 WWW Connection variable substitution scheme and macro file syntax can be found in References 13 and 15.

The main advantages in using macro files to establish connection between the Web server and databases are:

1. No significant coding effort or programming logic is necessary to build new applications.
2. The creator of the macro can use the full power

Figure 5 DB2 WWW Connection run time



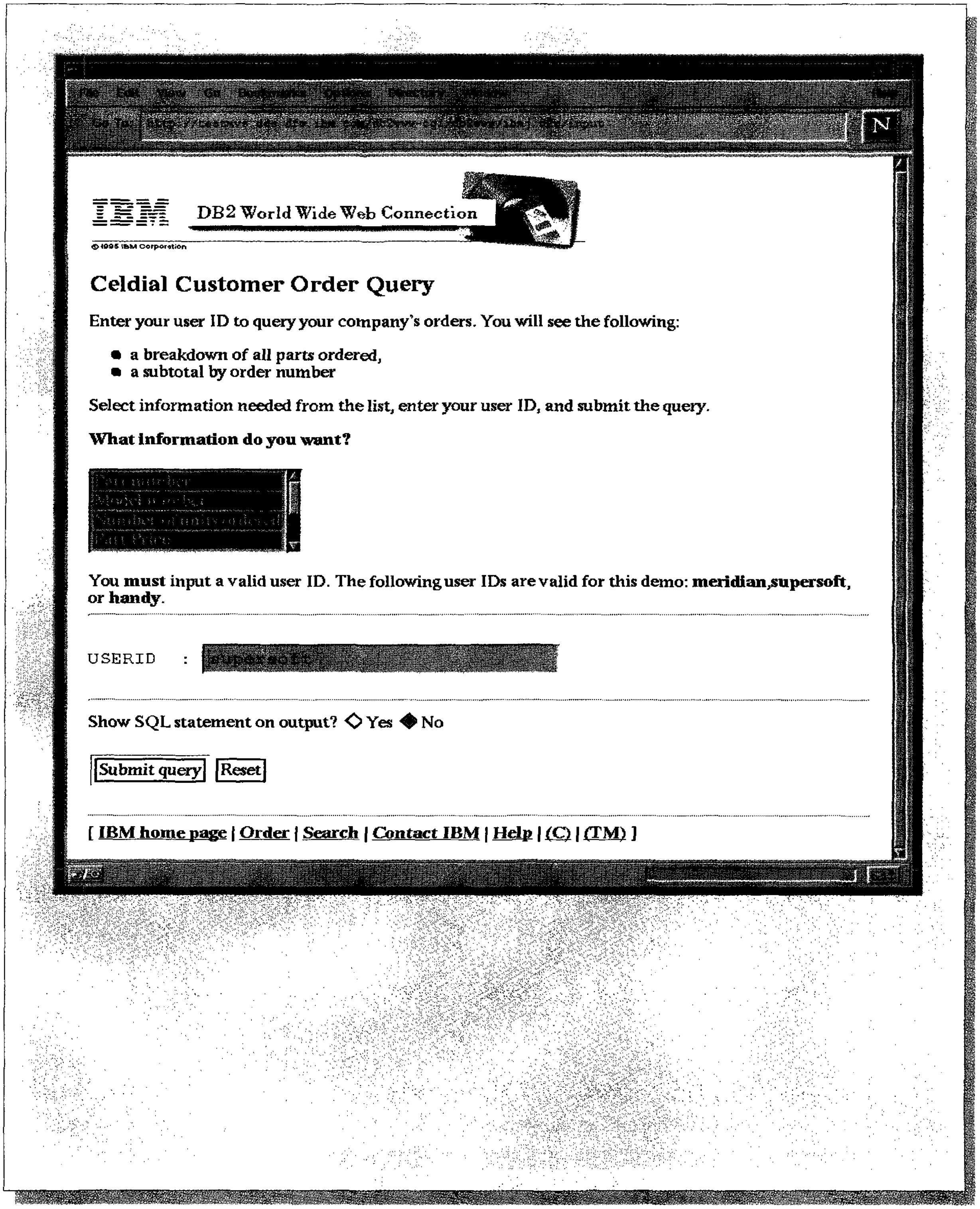
of the newest HTML version and SQL to develop macros.

3. Macros are easily portable to multiple server platforms since they are basically platform-independent. DB2 WWW Connection is an available product on all IBM and several popular non-IBM platforms. The product was used to create an online ticket sales application for the 1996 Olympics in Atlanta, Georgia, and is being used in several diverse Web applications involving database access, such as a "Yellow Pages" directory search, guest book sign-ups, travel information and reservations, campus directory service, job search services, and many others.

Net.Data. In this section, we discuss Net.Data, a successor to DB2 WWW Connection that greatly improves on both the functionality and performance of its predecessor. The basic structure of DB2 WWW Connection has been retained in Net.Data. The three major improvements are greatly expanded macro language features, support for multiple language environment backends apart from the DB2 backend, and performance improvements. Herein, the *backends* refer to pluggable components that are useful in database access, data manipulation, and table formatting. The *front ends*, on the other hand, denote the pre-process procedures whose primary functions include receiving requests from the Web servers, parsing the input and query strings, and invoking the suitable language backends. Figure 8 gives an overview of Net.Data. In the remainder of this section, we examine each of the three improvements in turn, and define some of the terminology used in Figure 8.

Several new enhancements have been made to the macro file syntax to aid in easier development and organization of macro files and also to allow for more flexible run-time processing. Macro files can now include other files, much in the same manner as the #include statement in some programming languages. Moreover, a new language feature allows files to be retrieved from Web servers and incorporated into the macro file at run time. Apart from the earlier HTML_INPUT, HTML_REPORT, and SQL sections that were supported by DB2 WWW Connection, macro file sections can now have names. The mode field, which is obtained from the URL, now can be used to specify a named macro section rather than just input and report sections. Also, a language feature has been provided to perform conditional execution of macro sections. This feature can be utilized to selectively execute certain sections at run time depending on user input. Details of the expanded macro language features and syntax may be found in the Net.Data reference and programming guides, which are downloadable from the Net.Data World Wide Web Internet site.[14]

Figure 6 Application input form


File Edit View Go Bookmarks Options Directory Window Help
Go To: http://[illegible]
N
IBM
DB2 World Wide Web Connection
© 1995 IBM Corporation

Celdial Customer Order Query

Enter your user ID to query your company's orders. You will see the following:

- a breakdown of all parts ordered,
- a subtotal by order number

Select information needed from the list, enter your user ID, and submit the query.

What information do you want?

Part number
Model number
Number of units ordered
Part Price

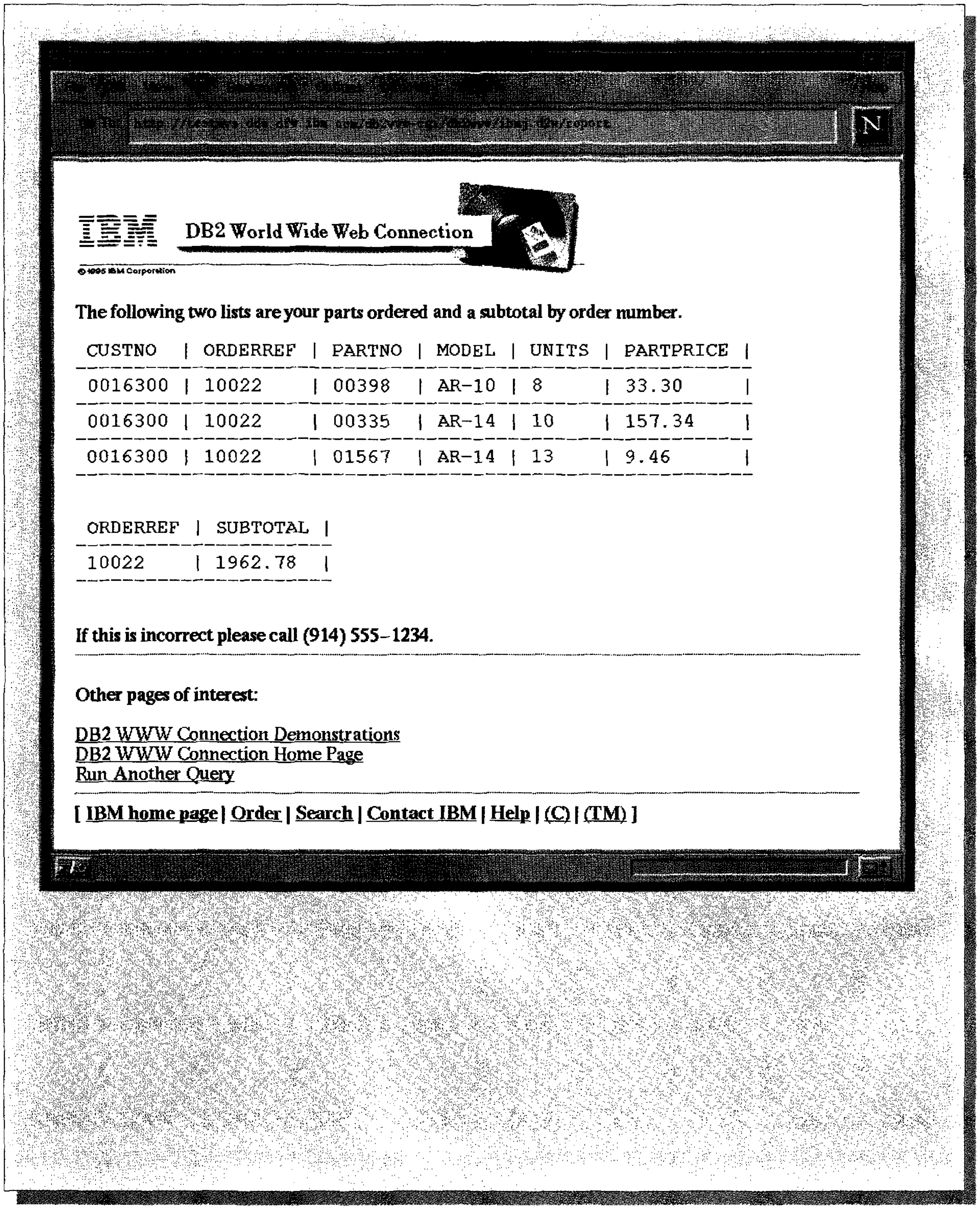
You **must** input a valid user ID. The following user IDs are valid for this demo: **meridian,supersoft**, or **handy**.

USERID : supersoft

Show SQL statement on output? ◇ Yes ◆ No

[IBM home page | Order | Search | Contact IBM | Help | (C) | (TM)]

Figure 7 Application report form



The following two lists are your parts ordered and a subtotal by order number.

```
CUSTNO   | ORDERREF  | PARTNO | MODEL | UNITS | PARTPRICE |
-----------------------------------------------------------
0016300 | 10022     | 00398  | AR-10 | 8     | 33.30     |
-----------------------------------------------------------
0016300 | 10022     | 00335  | AR-14 | 10    | 157.34    |
-----------------------------------------------------------
0016300 | 10022     | 01567  | AR-14 | 13    | 9.46      |
-----------------------------------------------------------

ORDERREF | SUBTOTAL |
----------------------
10022    | 1962.78  |
----------------------
```

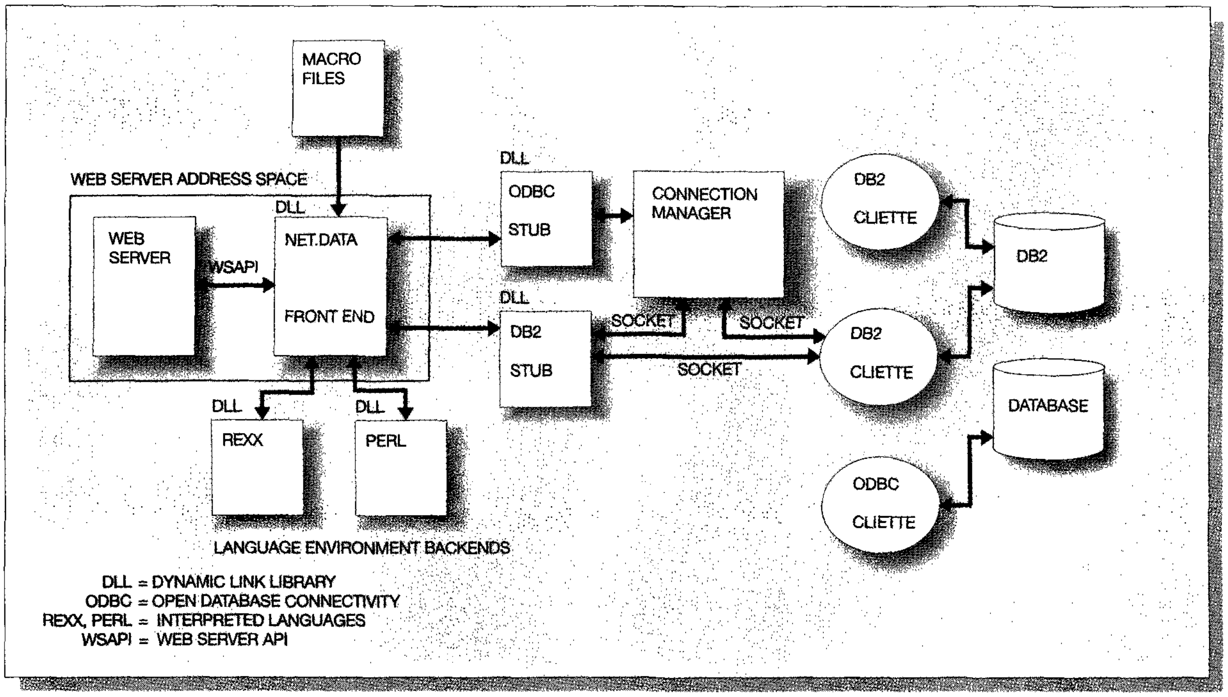
If this is incorrect please call (914) 555-1234.

Other pages of interest:

DB2 WWW Connection Demonstrations
DB2 WWW Connection Home Page
Run Another Query

[IBM home page | Order | Search | Contact IBM | Help | (C) | (TM)]

Figure 8 Net.Data overview

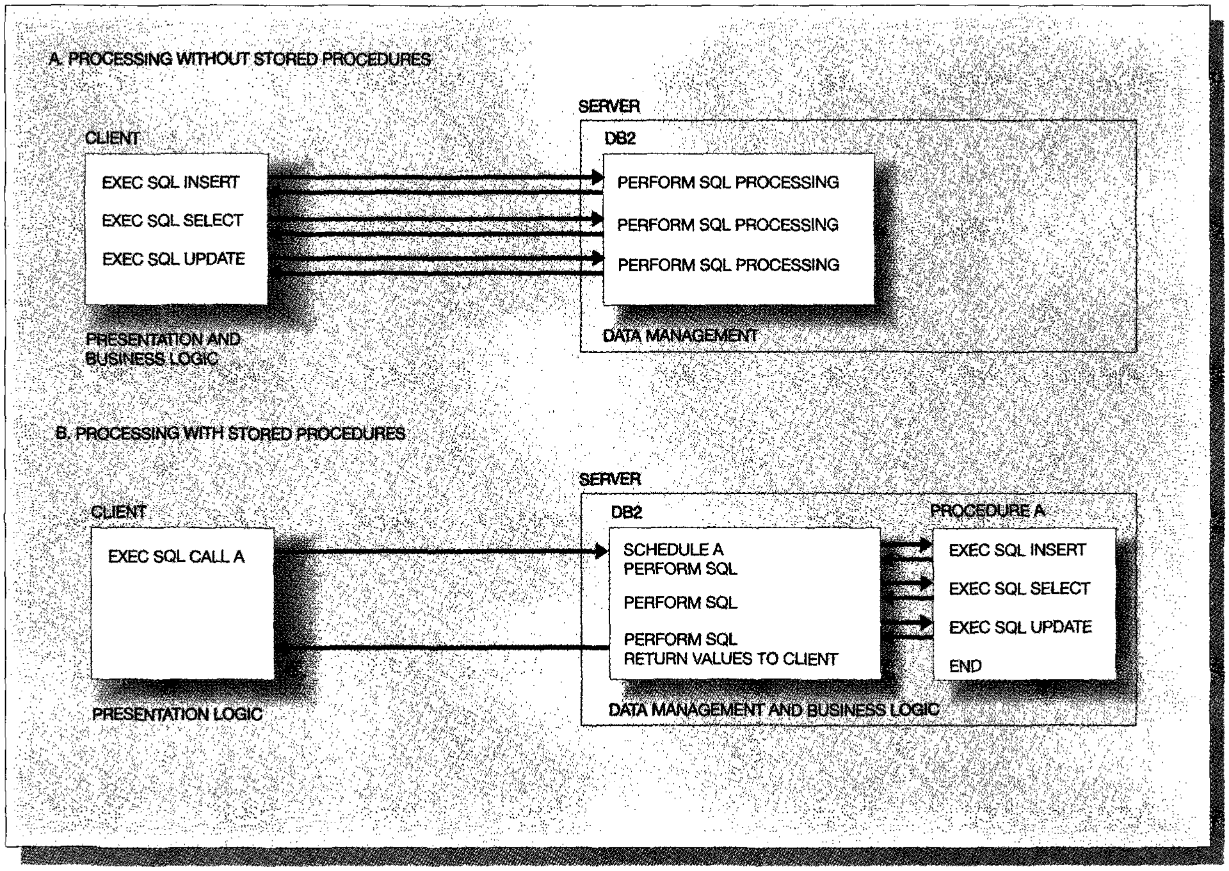


An important new feature that improves both the flexibility and performance of Net.Data is support for DB2 Stored Procedures.[16] Stored procedures are user-written programs that are stored at the DB2 server and can be invoked by client applications. They can be used to combine business logic and database access into a single procedure stored at the server. Stored procedures provide the advantage of application reuse, much in the same manner as library routines and performance improvements, especially in the situations where the client has to access a database at a remote location. Figure 9 gives an overview of processing with and without stored procedures.

Net.Data provides pluggable dynamic link libraries (DLL) for several language backend APIs. Inside the macro files, DTW_REXX, DTW_PERL, and DTW_SYSTEM define functions written in IBM's REXX (Restructured EXtended eXecutor) and Larry Wall's Perl (Practical Extraction and Report Language) interpreted languages, and compiled executable programs. REXX and Perl programs can be included either in line, or placed in external files that are invoked by %EXEC within the functions. The compiled executable programs must be invoked externally, and can be written in high-level languages, such as C and C++. The File backend, identified by DTW_FILE, allows the creation, deletion, and insertion of plain text files. Net.Data also defines many default C functions that can perform string manipulations, table formatting, and simple mathematics. These default functions have been provided to make the task of the Net.Data application developer easier. Figure 10 shows a Net.Data macro file that uses the same HTML_INPUT section as in Figure 4, but invokes additional REXX procedures to format the result tables in the HTML_REPORT section. The new report screen is shown in Figure 11.

All of the backend APIs include an externally exportable function, *execute( )*, which can parse input variables and values, invoke language interpreters (REXX and Perl) or shell commands (SYSTEM) to execute the individual language functions, and retrieve output variables and values. In general, input variables and values are passed into language functions by setting environment variables, while output variables and values are returned via named pipes or external files.

Figure 9 Processing with and without stored procedures



Major performance improvements in Net.Data have been achieved in two ways. The first is the introduction of a module called the connection manager.[17] A significant portion of the time spent in the DB2 backend in DB2 WWW Connection was due to the opening and closing of the connection to DB2. The connection manager is a long-running background process (i.e., *daemon*) that can be used to start up several processes, called *cliettes*, when it starts up. Cliettes continue running until the connection manager shuts down. The DB2 backend for Net.Data has been re-implemented as a cliette. DB2 cliettes open a connection to DB2 once they are started up. The connection remains open as long as the DB2 cliette is running. The original DB2 backend has been replaced by a stub routine. When an SQL statement needs to be processed by the DB2 backend, the Net.Data front end passes the request on to the DB2 stub. The stub routine obtains the address of a free DB2 cliette from the connection manager and passes on the request to the cliette, which then prepares the SQL, passes the request on to DB2 (reusing the already open connection), and passes the response back to the stub. The communication between connection manager, cliette, and stub is done through *sockets*, the standard Berkeley UNIX** end points of communications.

The second major performance improvement in Net.Data has been achieved by implementing the front end as a dynamic link library (DLL) that can be invoked via a Web server specific API. Supported APIs for the MVS platform are the IBM Internet Connection Secure Server's (ICSS) ICAPI and Netscape Server's NSAPI, although Microsoft's Internet Information Server's ISAPI is also supported on some other platforms. With the implementation of the front end as a DLL, multiple requests to Net.Data do not result in the spawning of separate address spaces to service each request. Instead, threads are created in the same address space on a per-request basis, with consequent performance improvements.

Net.Data is a generally available product on all IBM platforms, including MVS. Net.Data is being used by several customers to provide Web applications that involve database search. Airline reservations, library search, patent search, and news services are some of the business applications that have been developed around Net.Data. Some customers using Net.Data are Japan Airlines, The Boeing Company, Chrysler Corporation, and the Volvo Group. Some news items about how customers are actually using Net.Data may be retrieved from the Net.Data home page.[14] An example of Net.Data's use in corporate intranets is provided in the article "Boeing Gives Web Users First Class Access to DB2 Databases."[18] The DB2 Technical Services Organization had been providing timely, up-to-date information on DB2 announcements and product tips, upcoming training sessions, and many other related topics, through a news application that enabled users to access news items residing in DB2 databases. The original application was developed using Interactive System Productivity Facility (ISPF), an IBM software package that runs on IBM mainframes and thus requires a 3270-like terminal attached to the mainframe to be used. The application was rewritten using Net.Data, which proved an excellent means of marrying the intuitive and convenient browser interface with the vast data resources in their DB2 records. Users can now access the same information much more conveniently from any client platform that supports a browser.

Related work

In this section, we review solutions provided by industry vendors. The products discussed here may belong to one or two of the different categories we described earlier—Web server enhancements for DBMS support, Web-enabled DBMS, and pure middleware.

One of the popular commercial DBMS vendors, Oracle Corporation,[19] supplies Web connectivity with WebServer** 2.1. Oracle's WebServer contains a Web Request Broker (WRB), a high-performance multithreaded, multiprocess HTTP server; a WebServer SDK (Software Development Kit), delivering seamless integration with the Oracle7 database; and WebServer Management Tools, supporting HTML-based server management. The WebServer SDK fully encapsulates the WRB API and supports Java, PL/SQL**, LiveHTML**, and C++.

Sybase, Inc.[20] Enterprise CONNECT** middleware family enables corporations to provide an "any-to-any" computing relationship in today's complex computing environment. DirectCONNECT** provides the fundamental middleware building block to non-Sybase data. Mainframe APIs support linkage with applications on the mainframe (CICS, IMS Transaction Manager, MVS, and VSE); OmniCONNECT** offers users transparent read/write access to data across many heterogenous data sources. Web.sql** allows Web applications to dynamically link into corporate information systems to extend business functions. Web.sql also offers database gateway for DB2 with IBM Distributed Relational Database Architecture (DRDA*) communications protocol.

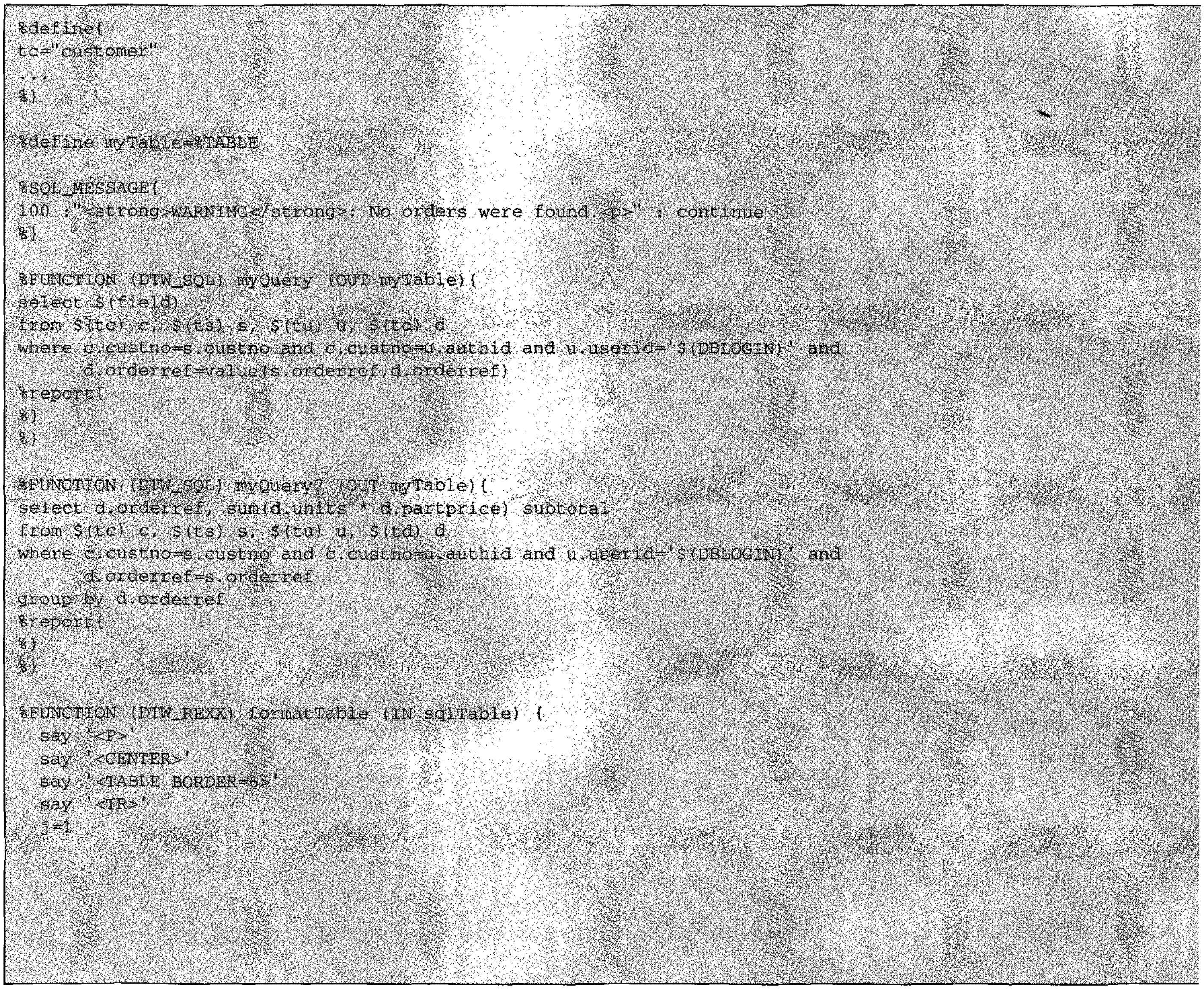
Informix's[12] INFORMIX-Universal Server** is an enterprise-capable and fully extensible relational DBMS, based on its core parallel database technology, Dynamic Scalable Architecture (DSA). Web connectivity can be achieved by using CGI interface kits.

The EDA** (Enterprise Data Access) middleware,[21] introduced by Information Builders Inc. in 1991, was conceived upon the concept that information delivery is most efficient when the user is shielded from the complexities of data file structures and communications networks. It provides cost-effective data integration solutions for on-line transaction processing (OLTP), data warehousing, and decision support systems.

Shadow Direct** from NEON System, Inc.[22] is an OLTP-oriented solution for DB2 access, including features like optimized ODBC** catalogs, dynamic-to-static SQL conversion, and data stream compression. The Shadow Direct allows popular ODBC-compliant desktop applications to run seamlessly and transparently with data retrieved from enterprise DB2 tables, IMS message queues and databases, existing CICS transactions, Virtual Storage Access Method (VSAM) files, and other data sources.

HeatShield** is XDB Systems'[23] approach to access databases from the Web. Connection between the Web user and the database can be established by using ODBC-enabled connection agents. Optimized access to DB2 can be achieved by ODBC static bind and catalog caching at run time. XDB also provides Java Enterprise Tools (JET) including JETConnect**, a

Figure 10 An example of a macro file (Net.Data)



```
%define{
tc="customer"
...
%}

%define myTable=%TABLE

%SQL_MESSAGE{
100 :"<strong>WARNING</strong>: No orders were found.<p>" : continue
%}

%FUNCTION (DTW_SQL) myQuery (OUT myTable){
select $(field)
from $(tc) c, $(ts) s, $(tu) u, $(td) d
where c.custno=s.custno and c.custno=u.authid and u.userid='$(DBLOGIN)' and
      d.orderref=value(s.orderref,d.orderref)
%report{
%}
%}

%FUNCTION (DTW_SQL) myQuery2 (OUT myTable){
select d.orderref, sum(d.units * d.partprice) subtotal
from $(tc) c, $(ts) s, $(tu) u, $(td) d
where c.custno=s.custno and c.custno=u.authid and u.userid='$(DBLOGIN)' and
      d.orderref=s.orderref
group by d.orderref
%report{
%}
%}

%FUNCTION (DTW_REXX) formatTable (IN sqlTable) {
  say '<P>'
  say '<CENTER>'
  say '<TABLE BORDER=6>'
  say '<TR>'
  j=1
```

DBMS-independent solution for database access from Java applications.

WebObjects** from NeXT Software, Inc. (acquired by Apple Computer, Inc.[24]) is a development environment that allows programmers to create connectivity directly between the Web and almost any relational and nonrelational DBMS.

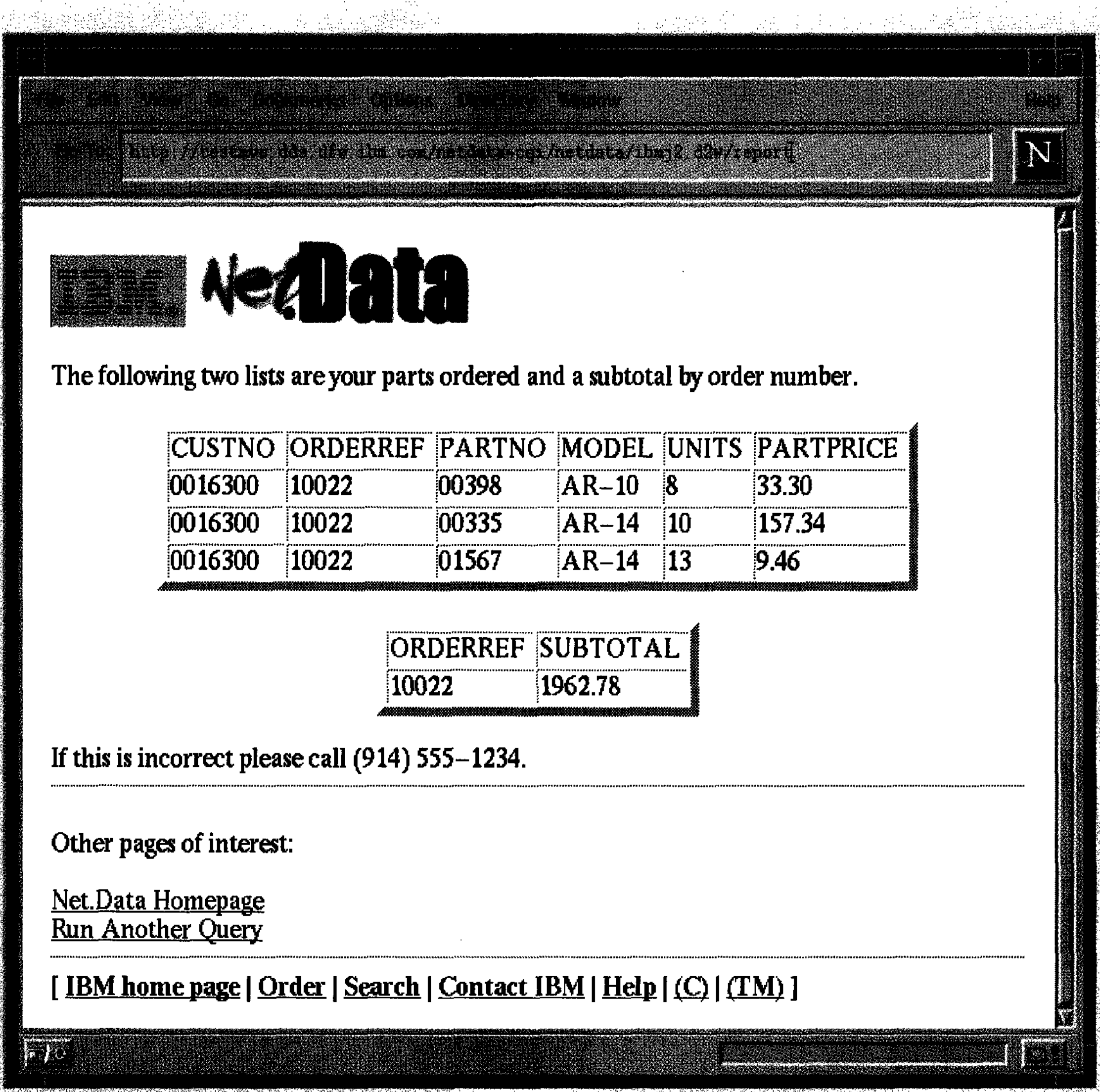
Microrim, Inc.[25] developed a product called R:WEB 1.0 that works with its own desktop relational database system R:BASE** 5.5. Based on the support for local area network (LAN) and SQL from R:BASE, R:WEB adds the functionality to convert R:BASE forms into HTML forms that interact with R:BASE database and ODBC data sources. Similarly, R:WEB also converts query results into HTML pages.

```
    DO until j > sqlTable_COLS
      say '<TD>'
      say sqlTable_N.j
      say '</TD>'
      j = j + 1
    end
    say '</TR>'
    i = i
    DO until i > sqlTable_ROWS
      say '<TR>'
      j = 1
      DO until j > sqlTable_COLS
        say '<TD>'
        say sqlTable_V.i.j
        say '</TD>'
        j = j + 1;
      end
      say '</TR>'
      i = i + 1
      end
        say '</TABLE>'
        say '</CENTER>'
%}

%HTML_INPUT{
...
%}

%HTML_REPORT{
<body bgcolor="#FFFFFF">
<TITLE>Net.Data Customer Order Results</TITLE><img src="/icons/netdata.gif">
<BR>
<P>The following two lists are your parts...
<BR>
@myQuery (myTable)
@formatTable (myTable)
@myQuery2 (myTable)
@formatTable (myTable)
<P>If this is incorrect please call (914) 555-1234.
<HR>
<P>Other pages of interest: ...
%}
```

Microsoft's[10] Internet Information Server (IIS) supports Microsoft's SQL Server and conforms to the ODBC standard. Microsoft also ships its own proprietary API (ISAPI) to let developers extend the server functionality for Web connectivity. In addition, Microsoft's SNA Server** provides a reliable and flexible platform for PC-to-host connectivity using advanced client/server architecture. The SNA servers can connect to the mainframe and IBM AS/400* systems using Systems Network Architecture (SNA) protocols.

Netscape's[11] Enterprise Server** 3.0 is a Web-enabled approach to DBMS connection. JavaScripts running on Enterprise Server 3.0 can transparently access any relational database system, including

Figure 11 Application report form (Net.Data)



The following two lists are your parts ordered and a subtotal by order number.

CUSTNO	ORDERREF	PARTNO	MODEL	UNITS	PARTPRICE
0016300	10022	00398	AR-10	8	33.30
0016300	10022	00335	AR-14	10	157.34
0016300	10022	01567	AR-14	13	9.46

ORDERREF	SUBTOTAL
10022	1962.78

If this is incorrect please call (914) 555-1234.

Other pages of interest:

Net.Data Homepage
Run Another Query

[IBM home page | Order | Search | Contact IBM | Help | (C) | (TM)]

CA/Ingres, Informix, Microsoft, Oracle, and Sybase databases. ODBC support is included as well. Netscape provides a lower-level Server API (NSAPI) to allow seamless integration of user-specified functions into the Netscape Enterprise Server. Currently, they are working with other vendors to define a cross-platform standard server interface using Java and JavaScript as well as cross-platform database accessibility using JDBC (Java Database Connectivity) technology.

Conclusions

Adding Web-based access to existing information technology structure is a cost-effective solution to enhance the efficiency and productivity of an organization. In this paper, we have discussed the fundamentals and application architecture to make existing business data ready for the Web. We have introduced IBM's novel middleware solutions for Web access to DB2 data, DB2 World Wide Web Connection, and Net.Data. Other industry solutions to the same problem on different platforms have also been surveyed. Unlike IBM's pure middleware approach, many of these solutions involve enhancements to the Web server or the DBMS.

Future directions for Net.Data include increasing the number of IBM-provided pluggable DLL backends (backends for executing Java applications on the server and for serving Java applets to the client are already provided on many platforms), providing an open API for loading and executing new backends, thus enabling the customers to plug in their own backends to Net.Data.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., Microsoft Corporation, Netscape Communications Corporation, Microrim, Inc., Informix Software, Inc., Information Builders Inc., Lotus Development Corp., X/Open Co., Ltd., Oracle Corporation, Sybase, Inc., NEON System, Inc., XDB Systems, Inc., or NeXT Software, Inc.

Cited references

1. The Common Gateway Interface, University of Illinois, Urbana-Champaign, IL, http://hoohoo.ncsa.uiuc.edu/cgi/ (1995), (home page: http://www.uiuc.edu).
2. Java Home Page, Sun Microsystems, Inc., 2550 Garcia Ave., Mountain View, CA 94043-1100, http://java.sun.com (home page: http://www.sun.com).
3. Network Computer, Inc., 500 Oracle Parkway, MS SC-2, Redwood Shores, CA 94065 (home page: http://www.nc.com).
4. IBM DB2 Universal Database Home Page, IBM Corporation, http://www.software.ibm.com/data/db2/udb/ (home page: http://www.ibm.com).
5. IBM CICS Transaction Processing Software Home Page, IBM Corporation, http://www.hursley.ibm.com/cics/ (home page: http://www.ibm.com).
6. IBM IMS Family Home Page, IBM Corporation, http://www.software.ibm.com/data/ims/ (home page: http://www.ibm.com).
7. Basic HTTP Protocol, World Wide Web Consortium, http://www.w3.org/Protocols/HTTP/HTTP2.html (home page: http://www.w3.org/.
8. NCSA—A Beginner's Guide to HTML Home Page, The National Center for Supercomputing Applications (NCSA), University of Illinois, Urbana-Champaign, IL, http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html (home page: http://www.uiuc.edu).
9. HTML Data Format, http://www.w3.org/MarkUp/MarkUp.html (home page: http://www.w3.org).
10. Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399, http://www.microsoft.com/iis/ (home page: http://microsoft.com).
11. Netscape Communications Corporation, 501 E. Middlefield Rd., Mountain View, CA 94043, http://www.netscape.com/comprod/server_central/product/enterprise/index.html (home page: http://home.netscape.com).
12. Informix Software, Inc., 4100 Bohannon Drive, Menlo Park, CA 94025, http://www.informix.com/informix/products/techbrfs/tbindx.htm (home page: http://www.informix.com).
13. DB2 WWW Connection Home Page, IBM Corporation, http://www.software.ibm.com/data/db2/db2wfac2.html (home page: http://www.ibm.com).
14. Net.Data Home Page, IBM Corporation, http://www.software.ibm.com/data/net.data (home page: http://www.ibm.com).
15. T. Nguyen and V. Srinivasan, "Accessing Relational Databases from the World Wide Web," *Proceedings of ACM SIGMOD* Conference (1996).
16. *Getting Started with DB2 Stored Procedures*, SG24-4693-00, IBM Corporation (1996); available through IBM branch offices.
17. Y.-H. Liu, P. Dantzig, C. E. Wu, and L. M. Ni, "A Distributed Connection Manager Interface for Web Services on IBM SP Systems," *Proceedings of International Conference on Parallel and Distributed Systems* (June 1996).
18. Data Management Customer Spotlight, IBM Corporation, http://www.software.ibm.com/data/solutions/customer/ boeing/boeing.htm (home page: http://www.ibm.com).
19. Oracle Corporation, 500 Oracle Parkway, P.O. Box 659506, Redwood Shores, CA 94065 (home page: http://www.oracle.com).
20. Sybase, Inc., 6475 Christie Ave., Emeryville, CA 94608, http://www.sybase.com/products/entcon/ (home page: http://www.sybase.com).
21. Information Builders Inc., 1250 Broadway, New York, NY 10001-3782, http://www.ibi.com/products.html (home page: http://www.ibi.com).
22. NEON System, Inc., 14141 Southwest Freeway, Suite 6200, Sugar Land, TX 77478, http://www.neonsys.com/sd_desk.htm (home page: http://www.neonsys.com).
23. XDB Systems, Inc., 9861 Broken Land Parkway, Columbia, MD 21046, http://www.xdb.com/hs/moredetail_hs.htm (home page: http://www.xdb.com).
24. NeXT Software, Inc., a subsidiary of Apple Computer, Inc., http://www.next.com/WebObjects/Mainframe_Brief.html (home page: http://www.next.com).

25. Microrim, Inc., 15395 S.E. 30th Place, Bellevue, WA 98007-9918, http://www.microrim.com/RBASE_Products/Software/RWeb_Features.html (home page: http://www.microrim.com).

Accepted for publication March 10, 1997.

Ching-Shan Peng *Department of Electrical and Computer Engineering, University of California, Irvine, California 92697 (electronic mail: cpeng@ece.uci.edu).* Mr. Peng worked at the IBM Thomas J. Watson Research Center, Hawthorne, New York, as a summer intern during 1996 and 1997. He received the B.S. degree in computer science and information engineering from National Chiao Tung University, Taiwan, in 1989 and M.S. degrees in both the department of electrical and computer engineering and the department of computer science from the University of Wisconsin, Madison, in 1993 and 1994, respectively. Mr. Peng is currently a Ph.D. candidate at the University of California, Irvine. His research interests include real-time scheduling, real-time database systems, and real-time application frameworks.

Shyh-Kwei Chen *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: skchen@watson.ibm.com).* Dr. Chen is a research staff member at the IBM Thomas J. Watson Research Center, Hawthorne, New York. He received the B.S. degree in computer science and information engineering from National Taiwan University in 1983, the M.S. degree in computer science from the University of Minnesota in 1987, and the Ph.D. degree in computer science from the University of Illinois, Urbana-Champaign, in 1994. His research interests include parallel processing, compilers, and fault-tolerant computing.

Jen-Yao Chung *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, New York 10598 (electronic mail: jychung@watson.ibm.com).* Dr. Chung is a research staff member at the IBM Thomas J. Watson Research Center, Hawthorne, New York. He currently is manager of the electronic commerce platforms department. He received the B.S. degree in computer science and information engineering from National Taiwan University, and M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign. His research interests include Web data access, network computing, scalable Web server, database performance and benchmark, and parallel database/query systems. Dr. Chung has won one IEEE Outstanding Paper Award, an IBM Technical Achievement Award, a Research Division Technical Group Award, and a Research Division Award. He has served as industrial chair, program committee member, and session chair in several workshops and conferences. Dr. Chung is a senior member of IEEE and a member of ACM.

Amber Roy-Chowdhury *Transarc Corporation, The Gulf Tower, 707 Grant Street, Pittsburgh, Pennsylvania 15219 (electronic mail: Amber_Roy-Chowdhury at transarc.com).* Dr. Roy-Chowdhury is currently employed by Transarc Corporation, a subsidiary of IBM. He was employed by IBM at the Thomas J. Watson Research Center in the summers of 1992 and 1993 in the fault-tolerant systems department and from December 1995 to May 1997 in the data intensive platforms department. Dr. Roy-Chowdhury received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kharagpur, and M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1990, 1992, and 1995, respectively. His research interests include fault-tolerant computer systems, parallel and distributed computing, and transaction processing. He is a member of ACM, IEEE, and the IEEE Computer Society.

V. Srinivasan *Network Computer, Inc., 500 Oracle Parkway, MS SC-2, Redwood Shores, California 94065 (electronic mail: srini@navio. com).* Dr. Srinivasan is currently a senior technical staff member at Network Computer, Inc. (formerly Navio Communications, Inc.). At Network Computer and Navio, he has contributed extensively to the design and implementation of the NCTV Custom Connect Server, which provides highly scalable access to data for enhanced TV clients. Prior to joining Navio, Dr. Srinivasan was a member of the Database Technology Institute at the IBM Santa Teresa Laboratory. At IBM, he played key roles as an architect and implementor of several products including Net.Data, DB2 WWW Connection, ObjectLite, and ObjectStore/DB2 gateway. Based on his work at IBM and Navio Communications, Dr. Srinivasan has filed for several patents in the areas of the Internet and object technology. He received his B.Tech. degree in computer science from the Indian Institute of Technology, Madras, and his M.S. and Ph.D. degrees in computer science from the University of Wisconsin, Madison.

Reprint Order No. G321-5667.