PAPER Special Issue on Knowledge-Based Software Engineering

Internet/Intranet Application Development System WebBASE and Its Evaluation

Shuichiro YAMAMOTO[†], *Member*, Ryuji KAWASAKI[†], Toshihiro MOTODA[†], *and* Koji TOKUMARU[†], *Nonmembers*

SUMMARY There is increasing demand for corporate information systems that have a simple human interface and are easy to access via WWW browsers. This paper proposes WebBASE, which integrates the WWW and relational databases. Experimental evaluation shows that WebBASE offers superior performance compared to existing products. Field studies of actual WebBASE applications show that it can improve the productivity of software developers for intranet application development.

key words: *WWW, internet, intranet, middleware, distributed information system, WWW-database connectivity*

1. Introduction

The expanding Internet environment has made the WWW more popular. As a result, more users are accessing information servers world-wide. Also, in the process of establishing Electronic Commerce, such as on-line shopping, it has become essential to integrate databases to the WWW. More and more companies would like to develop their own intra-company information systems that allow access via the WWW, because such systems are inexpensive, can operate on various operating systems and computers, and are easy to use.

When setting up WWW servers to provide public information, information pages must be made using the Hyper Text Markup Language (HTML). These pages must be frequently updated, and this is best achieved by integrating databases to the WWW. Integrating the WWW with databases has so far required the development of applications written in C Language or by shell scripts on UNIX in accordance with a Common Gateway Interface (CGI) defined by the WWW server. Our experiments in integrating the WWW to databases via gateway applications based on CGI have shown that we have the following problems [1].

(1) Process execution by CGI is necessary, and opening and closing a database for each process executed degrades access performance.

(2) Because WWW protocols require that connections and disconnections be made every time a page is accessed, the protocols cannot control database transactions during a series of accesses to multiple pages.

(3) Application development becomes inefficient because the database and WWW information pages must coordinate for each transaction using C Language constructs or shell scripts.

To solve these problems, we have developed WebBASE. This has an extended HTML script language and session processing, which can execute programs described by SQL and HTML. WebBASE is based on VGUIDE [2], which is middleware for developing client/server systems, and works with popular database products. Because it provides an interface to the WWW without CGI, it provides faster processing and better response even over heavily loaded networks. Also, WebBASE scripts reduce service development costs because they allows us to describe service transactions by HTML and SQL. This paper introduces WebBASE and evaluates its effectiveness based on experiments and questionnaires completed by intranet application developers.

2. Overview of WebBASE

2.1 System Configuration of WebBASE

Figure 1 shows the system configuration of WebBASE. WebBASE processes consist of the HTTP communication part, the WebBASE script interpreter, and the VGUIDE communication part. Between the WWW browser and the WebBASE, HTTP (Hyper Text Transfer Protocol) is used as the communication protocol. WebBASE can work in the same way as popular WWW servers, such as CERN, NCSA, and Netscape communication servers. WebBASE works with VGUIDE and DBMS on the same machine. From the viewpoint of security and distributed processing, they can also work on different machines. WebBASE generates database queries based on SQL statements in WebBASEscripts. Between WebBASE and VGUIDE, the VGUIDE protocol makes requests to execute SQL statements to acquire the results desired, then makes Remote Procedure Call (RPC). VGUIDE is middleware that is used to develop client/server information systems. It offers real-time transaction control by dispatching query requests for resident database processes; this can be applied to large-scale systems with heavily loaded networks.

Manuscript received April 15, 1998.
Manuscript revised July 15, 1998.

[†]The authors are with the NTT Multi-Media System Laboratory Group, Yokosuka-shi, 239–0847 Japan.

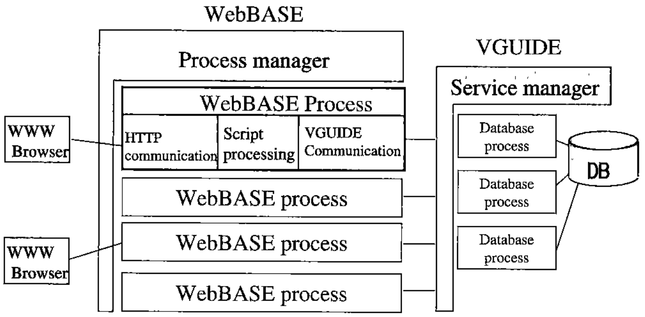


Fig. 1 WebBASE system configuration.

WebBASE allows the following functions by using VGUIDE as a database server.

(1) Users can access existing databases for major business transactions via the WWW because VGUIDE provides an open environment independent of the DBMS used.

(2) The Transaction Processing (TP) monitor preserves necessary resources before accessing a database to respond faster. WWW services can be easily added to existing major business transactions because VGUIDE puts less load on the database servers.

(3) WebBASE does not need to deal with the differences between DBMSs because VGUIDE handles them.

2.2 WebBASE Script

In this paper, we propose a new script language, named WebBASE script, to integrate HTML and SQL. As wrapper applications encapsulating differences of servers can be easily developed in WebBASE script, it is easy to integrate heterogeneous DBMS servers. WebBASE provides a new scripting language for wrapper-based heterogeneous server integration.

The components of the WebBASE script language are as follows.

(a) HTML for display on a monitor

(b) SQL statements for database access

(c) Processing statements for VBA compatibility

(d) Dynamic Link Library (DLL)

WebBASE scripts for database access have the twelve basic statements shown in Table 1. A dot ('.') at the head of extended parts (b), (c) above, distinguishes HTML statements (a) from the extended parts in the WebBASE Script.

An example of a WebBASE script is shown in Fig. 2. A complete language description of WebBASE is described in the authors' home page[3].

WebBASE also provides DLL functions. The DLL libraries are linked with WebBASE processes in advance and executed dynamically when they are called in script statements. DLL functions are used to develop wrapper applications that facilitate integration of heterogeneous servers.

2.3 Session Management Algorithm of WebBASE

WebBASE manages database access sessions over HTTP, basically a stateless protocol. The session management algorithm of WebBASE uses port numbers to identify sessions of the corresponding WWW browser. WebBASE manages resident processes for database access sessions on the server as shown in Fig. 1. The resident process reduces overhead in the initiation and termination process of CGI. When one WebBASE process is assigned for a request from a WWW browser, a port number is sent back to the WWW browser. Thereafter, the WWW browser can continue the database access session by sending requests tagged with the port number. WebBASE identifies the WWW browser by the port number and dispatches the requests from the WWW browser to the corresponding database access process without redundant overhead. The detail of the algorithm is as follows (Fig. 3).

(1) The WebBASE process, after receiving a request from a WWW browser, begins a session. The request includes a template file name and parameters. WebBASE processes the template file which is described in WebBASE script.

(2) WebBASE creates a unique identifier and sends it back to the WWW browser. The WWW browser makes requests to the WebBASE, which is running on the server according to the identifier (the port number and unique data) passed through the process manager.

(3) WebBASE reads and processes the WebBASE script described on the template.

  (a) Assigns parameters given at execution to variables

  (b) Connects to the database server (VGUIDE)

(4) If the statement is SQL, it requests the database server to begin a process.

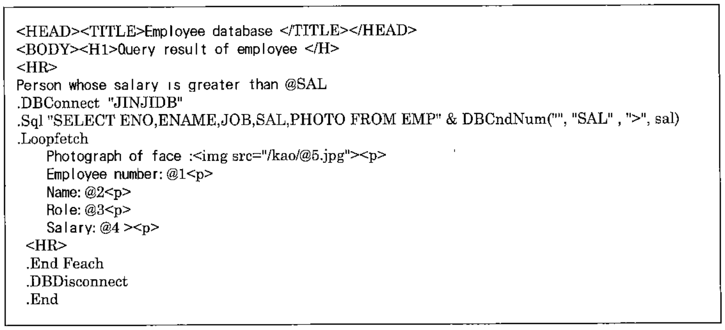
(5) WebBASE acquires the retrieval results from the database server, assigns acquired data to variables and combine the results into an HTML template. At this time, the retrieval results are assigned to variables @1, @2, ... based on the order of the columns.

(6) WebBASE converts combined data into HTTP and

Table 1 Overview of WebBASE script statements.

	Function	Statements
1	connection of database	.DBConnect <database name>
2	disconnection of database	.DBDisconnect
3	execution of SQL statements	.Sql <SQLstatements>
4	get query results by value	.LoopFetch (*) <a sequence of arguments> .End Fetch
5	get query result by string	. <variable name> = DBgetcol(<column number>)
6	write query result on a file	.DBWriteFile <file name>
7	remote procedure call	.DBCpexec <CP name > {, <argument list>}
8	get the result of R P C	.Dbdownload
9	conditional branch statement	.If <condition> Then <process block> .End If
10	assignment statement	.Let <variable name> = <value>
11	execution of external command	` <command name>
12	termination of session	.ExitSession

(*) < a sequence of arguments > is as follows.
<display name>: @<column number> {, <display name>: @< column number> }
Where display names are the field names on information pages.
The results of query are displayed as the value of the fields corresponding colums.



```
<HEAD><TITLE>Employee database </TITLE></HEAD>
<BODY><H1>Query result of employee </H>
<HR>
Person whose salary is greater than @SAL
.DBConnect "JINJIDB"
.Sql "SELECT ENO,ENAME,JOB,SAL,PHOTO FROM EMP" & DBCndNum("", "SAL" , ">", sal)
.Loopfetch
    Photograph of face :<img src="/kao/@5.jpg"><p>
    Employee number: @1<p>
    Name: @2<p>
    Role: @3<p>
    Salary: @4 ><p>
 <HR>
 .End Feach
 .DBDisconnect
 .End
```

Fig. 2 Example of WebBASE script.

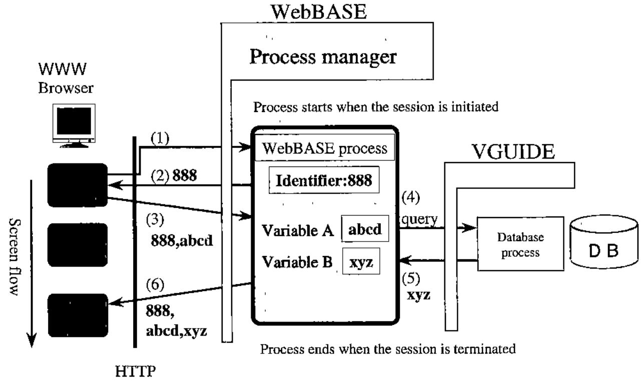


Fig. 3 Example of WebBASE session management.

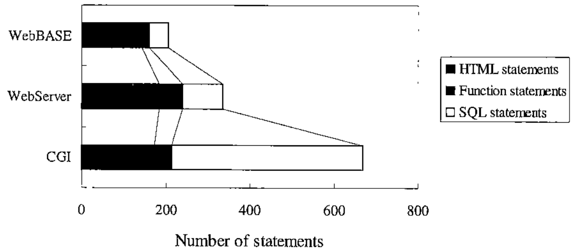


Fig. 4 Comparison of volumes.

transfers it together with its identifier to the browser via the process manager.

(7) When the EXITSESSION command in the template file is given, the WebBASE process is completed.

3. Experimental Evaluation of WebBASE

3.1 Amount of Description

Figure 4 compares the number of description lines needed with the CGI, WebServer and WebBASE approaches for the same application. The number of lines in WebBASE script is about 30.6% that in the CGI program codes for the same application. In detail, WebBASE reduces the numbers of lines for SQL, functional

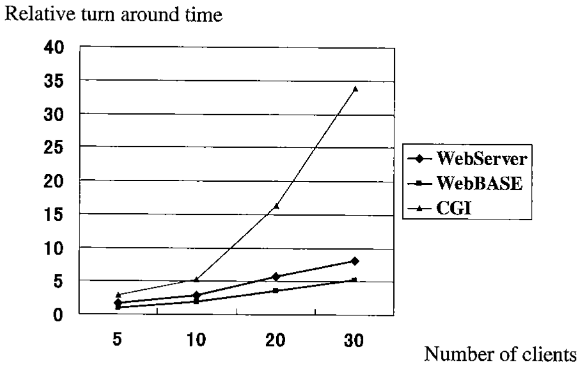


Fig. 5 Comparison of turn around time.

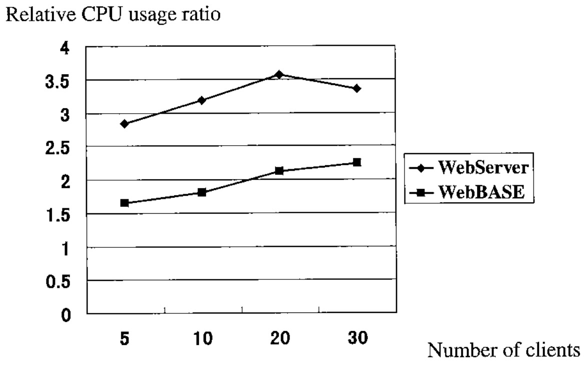


Fig. 6 Comparison of CPU usage ratio.

and HTML statements by 9.7%, 54.8%, and 18%, respectively, from those required by the CGI program codes. This figure also shows that WebBASE reduces the codes of the WebServer by 39% for the same application. This is done by integrating HTML and SQL statements in WebBASE script processing. This result shows the high productivity of WebBASE because it uses fewer statements.

3.2 Transaction Response Time

Experimental evaluations were conducted to compare the response time for a transaction using the WebBASE, CGI and WebServer approaches. For the comparison, the time required from issuing one query request to accepting the resulting HTML page were measured. The response times were measured for different numbers of simultaneous clients, ranging from 5 to 30. Experimental conditions are as follows.

(1) A tour database consists of a tour table and a price table. The tour record has 7 columns and occupies 92 bytes in total. The price table has 5 columns and occupies 44 bytes in total.

(2) The retrieve condition is to select appropriate records under a specified budget and display the contents of the retrieved records.

(3) Ten records were retrieved from the 300,000 records in the database.

(4) CGI is described in Perl script.

(5) Three SPARC Station 20 were used as the client, server, and database machine.

(6) 10 BASE-T was used as the network.

(7) WebServer Release 2.0 was used.

The result is shown in Fig. 5. Response time for a transaction using WebBASE is less than one fifth that using the CGI approach. Also the response time using WebBASE is approximately 59% of that using Webserver. This proves that the resident process control provided by WebBASE offers high performance. A comparison of CPU usage ratio is shown in Fig. 6. The CPU usage ratio of WebBASE is about 43% smaller than that of WebServer.

3.3 Evaluation by Questionnaire

A series of questions were used to evaluate the applicability of WebBASE for actual intranet development projects. The questionnaire was completed by managers of WebBASE application projects.

(1) The following four WebBASE application projects were investigated. These systems are all currently being used to provide actual services.

S1: Service order system. The volume of the system is 16 k lines of code. The development period was 8 months.
S2: Know-how sharing system. The volume of the system is 8 k lines of code. The development period was 2 months.
S3: Service order system. The volume of the system is 50 k lines of code. The development period was 8 months.
S4: Service order system. The volume of the system is 21 k lines of code. The development period was 2 months.

(2) Contents of questionnaire

The questionnaire contained 8 questions.
[Q1] How long was the development period using WebBASE in comparison to traditional client/server application development tools?
[Q2] How many lines of codes were there when using WebBASE in comparison to traditional client/server application development tools?
[Q3] How many problems were there when using WebBASE in comparison to traditional client/server application development tools?

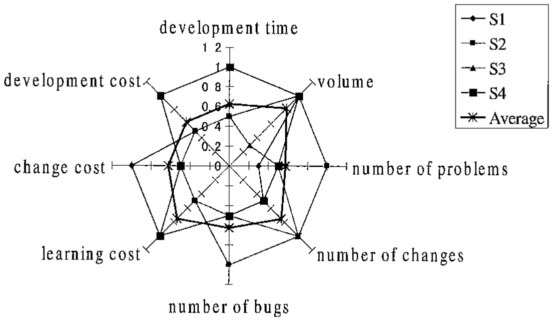


Fig. 7 Comparison of WebBASE application development with client/server application development.

[Q4] How many specification changes were there when using WebBASE in comparison to traditional client/server application development tools?

[Q5] How many bugs were there when using WebBASE in comparison to traditional client/server application development tools?

[Q6] How high was the learning cost when using WebBASE in comparison to traditional client/server application development tools?

[Q7] How high was the modification cost when using WebBASE in comparison to traditional client/server application development tools?

[Q8] How much the development manpower was needed when using WebBASE in comparison to traditional client/server application development tools?

(3) Questionnaire results

Figure 7 shows the questionnaire results for the four WebBASE application projects. On average, the productivity of WebBASE was approximately 1.3 times greater that of client/server application development tools. WebBASE cut the number of bugs per line of statements by 25% in comparison to client/server application development tools.

4. Considerations

4.1 Applicability

(1) Presentation

Distributed applications can be divided into presentation, function, and data tiers. WebBASE applications are constructed from WWW Browsers (presentation tier), WebBASE (function tier), and DBMS (data tier). If the requirements for presenting internet/intranet applications can be covered by a WWW browser, it is easy to develop a intranet application because WWW browsers can cut the development time due to their presentation portions; the function tier can be developed efficiently using WebBASE scripts. However, if the requirements of the presentation tier exceed the capabilities of WWW browsers, it is necessary to enrich them by developing relatively simple presentation applications, e.g., Java applets. In this case, Java DataBase Connectivity (JDBC) may be used on the client side. However, direct JDBC access through internet/intranet has been inhibited from the viewpoint of security. Therefore, JDBC should be used on the server side[4]. As Java applications on the server are called from the WWW server at the request of applets, there may be a redundant overhead for calling and terminating Java applications. On the other hand, there is no such redundant overhead in case of WebBASE. A quantitative evaluation comparing WebBASE and JDBC needs future research.

In case of actual business applications, complex form entry interfaces may be needed for business transaction processes. If the requirements for such presentations can not be implemented using WWW browsers, it is difficult to apply WebBASE because the presentation applications would not be available on the WWW browser.

(2) Reusability

In the case of WebBASE applications, it is necessary to create a WebBASE script file for each information page. This may increase the number of lines of WebBASE script statements, because similar information pages need similar lines of WebBASE script statements in different script files. To reduce the duplication of similar codes, the following macro statement is introduced to WebBASE script.

.source "< macro file name >"

Contents of the specified macro file are dynamically expanded into the script files that reference the macro file. The macro function reduces the total volume of WebBASE applications by defining a macro for reusing WebBASE script codes.

(3) Data type extension

WebBASE provides the array type, but does not provide complex data structures, such as, record, list, and pointer type. The reason why it does not allow complex data types is to provide a simple syntax and the efficient processing of WebBASE statements. If there is a need to manage complex data structures, an external DLL function managing the data could be defined and the statements calling them described in WebBASE script files. Thus, DLL functions could be used to extend the simple data structure of WebBASE.

(4) Port number

WebBASE issues a port number for each connection request. This may cause a communication problem, users fail to communicate with WebBASE, if their firewall restricts the range of port numbers. To solve this problem, we provide the port number range definition function. WebBASE generates port numbers so that their ranges fall within the defined ranges of port numbers.

(5) Access control

It is important to control information access according to the roles of users for intranet applications. Macros can be used for this purposes. A role model is defined that consists of organizations, roles, person, Web pages, and scopes. When user requests a Web page, the access control macro statements of the requested Web page check if the role of the user allows access to the Web page by retrieving the role model. If the check succeeds, the Web page will be displayed to the user, otherwise the request is rejected. The role model approach is robust because it is easy to modify records of the role model to reflect changes in the organization.

(6) Real examples

So far more than 200 internet/intranet applications have been developed by WebBASE. NTT DIRECTORY[5] and Surface Analysis Network Database[6] are examples of WebBASE applications. NTT DIRECTORY manages approximately 300 thousands Japanese home pages. It is well known as a major Japanese search engine. The number of hits on NTT DIRECTORY is about one million per day. This shows the effectiveness of WebBASE. Surface Analysis Network Database generates Web pages that include spectrum graphs of materials are generated on the series of numerical spectrum data stored in the database.

4.2 Points for Developing Applications

(1) Architectural design

WebBASE allows various types of client/server systems to be constructed. If a client function is made available using a WWW browser and WebBASE, and the client refers to the database by remote SQL using a WebBASE script, another new type of three-tiered model can be achieved which puts a part of the presentation function on the server. An important characteristic of this model is that the presentation function can be developed very easily by creating hypertext in HTML; in other words by creating a document, not by programming with existing GUI products. For a business information management system, the three-tiered model using the WWW server makes data available through existing information systems. On its data tier, only the necessary data is picked up and put in the UNIX database from the legacy system on the host computer. On the function tier, WebBASE script creates business applications using WebBASE. On the presentation tier, user interfaces independent of the operating systems of clients can be created using Netscape and EXCEL. Even if the WWW is used, the three-tiered model, which is becoming common among recent client/server systems, can make maintenance work more effective because this work can easily be expanded and more work can be added by modifying and/or adding templates. When a command procedure (CP), which is the fourth generation language (4GL) of VGUIDE, is executed through a WebBASE script, the common process logic described by CP can be reused. In practice, creating a script on template files is usually easier, and it is better from the viewpoint of quality and maintenance to create command procedures (CP) in VGUIDE and to describe execute RPC commands on WebBASE when multiple SQL statements must be issued, for example, when putting values input through the WWW browser into multiple tables in the database. In this case, the model should be called a four-tiered model because the server has the presentation tier and the function tier as different tiers. Installing applications such as Excel and Java with the WWW browser allow the installation of more customized distributed systems. In particular, with a WWW server, it is difficult to install systems allowing a huge number of clients to simultaneously connect to the server. This is because distributed loads, which are a characteristic of the client/server system, are not an advantage unless the clients perform some of the processes. Combining the existing three-tiered model and four-tiered model could solve this problem. In either case, the combination of VGUIDE and WebBASE is suitable for various kinds of client/server systems. From now, establishing a standard for choosing the most appropriate system architecture will become more important.

(2) Requirement analysis for the presentation tier

User interface requirements of intranets are likely to change more frequently than usual client server applications because prototyping intranet user interfaces is very easy thanks to HTML. This may cause the problem that it is difficult to fix the user interface specification early enough. If the new data items are added to an information page, the data model of the intranet may be impacted. This would affect the functions of other information pages because of the data model modifications. To solve this problem, the user interface specification must be fixed as early as possible.

(3) Number of processes

WebBASE executes applications efficiently by performing multiple processes simultaneously. It is necessary to determine the maximum number of processes that can run in parallel. Let W be the number of WebBASE processes, N the length of VGUIDE request queue, and V the number of VGUIDE processes. If $W > N + V$ then database processing will be over loaded. If $W < N + V$ then the database resources will not be fully used. Thus W should equal the sum of N and V for the efficient use of database resources.

4.3 Related Works

Related work on middleware that connects WWW to DBMS falls into three groups: work on generating HTML query forms through flexible user interface[7], work on reducing the mismatch between data manipulation languages (DML) such as SQL and HTML[8] and work on integrating heterogeneous DBMS servers on networks. The field of heterogeneous server integration includes the knowledge-based approach[9], wrapper-based approach[10] and catalogue-based approach[11] to manage different kinds of DBMS structures and operations to access DBMS servers. WebBASE script can be viewed as a script language for wrapping heterogeneous servers.

There are two kinds of approaches to reducing the mismatch between DML and HTML. The template approach provides an extended script language in which users can describe HTML statements, SQL statements, and control statements. The library approach provides library functions that generate HTML statements. Internet Database Connectivity (IDC)[12] of Microsoft and WebBASE are examples of the template approach. The productivity of the template approach is very high because HTML statements for the result of database queries are dynamically generated based on the script languages. In case of IDC, the OS is restricted to Windows, whereas WebBASE can work on Windows as well as UNIX. Thus, the scalability of WebBASE is greater than that of IDC.

WebServer[13] of Oracle and LiveWire Professional[14] of Netscape are examples of library approaches. WebServer provides Hyper Text Functions (HTF) generating HTML tags and Hyper Text Procedures (HTP) generating HTML statements for the results of database queries. Both HTF and HTP are provided for describing PL/SQL statements. LiveWire Professional provides database access interfaces as the external functions of Java scripts. As the library approaches use existing programming languages, they are not easier to use than the template approaches. User should also study the library functions provided. As the target database of WebServer is restricted to Oracle, it is difficult to apply it to multi-vendor intranets including different kinds of database management systems. WebBASE can work with Oracle, Sybase, and Informix, thanks to VGUIDE which encapsulates the differences of these database systems.

5. Conclusion

This paper has described the architectures, language features, and functionalities of WebBASE. This paper also demonstrated that WebBASE can reduce the volume and the number of bugs and the design phase problems in developing intranet applications. An experiment showed that the response time of WebBASE is superior than those of CGI and commercial products. The application of WebBASE to developing intranet systems was also described. The applicability of WebBASE to client server information systems based on the multi-tiered architecture will be evaluated.

References

[1] K. Tokumaru, T. Motoda, and H. Kurokawa, "Comparative study of WWW interfaces to relational databases," NTT REVIEW, vol.8, no.3, pp.74–79, 1996.
[2] S. Yamamoto, R. Kawasaki, and M. Nagaoka, "VGUIDE: 4GL application platform for large distributed information systems," COMPSAC'96, pp.536–541, 1996.
[3] WebBASE, http://robin.sl.cae.ntt.co.jp/vgindex.html
[4] E. Horowitz, "Migrating software to the World Wide Web," IEEE Software, vol.15, no.3, pp.18–21, 1998.
[5] NTT DIRECTORY: http://navi.ntt.co.jp
[6] K. Yoshihara and M. Yoshitake, "Construction of the surface analysis network database," Journal of Surface Analysis, vol.1, no.3, pp.369–373, 1995.
[7] S.P. Hadjiefthymiades and D.I. Martakos, "A generic framework for the deployment of structured databases on the World Wide Web," Proc. Fifth Int'l World Wide Web Conf., May 1996, http://www5conf.inria.fr/.
[8] T. Nguyen and V. Srinivasan, "Accessing relational databases from the world wide web," Proc. ACM SIGMOD Int'l Conf. On Management of Data, ACM Press, New York, pp.529–540, June 1996.
[9] M.R. Genesereth, A.M. Keller, and O.M. Duschka, "Infomaster: An information integration system," SIGMOD'97, pp.539–542, 1997.
[10] S.-T. Wi, T. Ohmori, and M. Hoshi, "An information integration architecture for mobile users in WWW environment," Trans. Information Processing Society of Japan, vol.39, no.4, pp.888–900, 1998.
[11] R.J. Miller, O.G. Tsatalos, and J.H. Williams, "DataWeb: Customizable Database Publishing for the Web," IEEE Multimedia, vol.4, no.4, pp.14–21, 1997.
[12] IIS, http://www.microsoft.com
[13] WebServer, http://www.oracle.com/
[14] Live Wire, http://home.netscape.com

Shuichiro Yamamoto is a senior research engineer, supervisor, of NTT Multi-Media System Laboratory Group. Previously, he was engaged in the development of CASE tools and distributed application development platforms. His research interests include distributed information systems, end user computing, reengineering, and requirements engineering. Mr. Yamamoto received a B.S. in information engineering from Nagoya Institute of Technology in 1977, and an M.S. in information engineering from Nagoya University in 1979. He is a member of IEEE, ACM, IPSJ and JSAI.

Ryuji Kawasaki is a senior research engineer, supervisor, of NTT Software Laboratories. Previously, he was engaged in the development of database middleware and virtual machine systems. His research interests include distributed information systems and end user computing. Mr. Kawasaki received a B.S. in electronical engineering from Nagoya Institute of Technology in 1979. He is a member of IPSJ.

Toshihiro Motoda is a senior research engineer of NTT Software Laboratories. Previously, he was engaged in the development of end user computing environments. His research interests include distributed information systems and end user computing. Mr. Motoda received a B.S. and M.S. in information engineering from Toyohashi University of Technology in 1987 and 1989. He is a member of IPSJ.

Koji Tokumaru is a manager, of NTT Software Corporation. Previously, he was engaged in the development of distributed application development platforms. His research interests include distributed information systems, end user computing and network. Mr. Tokumaru received a bachelor of engineering in electrical engineering from Kagoshima University in 1986. He is a member of IPSJ.