Databases on the Web: Technologies for Federation Architectures and Case Studies

Ralf Kramer
Forschungszentrum Informatik (FZI)
Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany
kramer@fzi.de; http://www.fzi.de/kramer.html

1 Introduction

The Internet and especially the World-Wide Web (WWW) provide a promising approach to make all kinds of data and information available on both the Internet and on private networks, so-called Intranets. However, making information available as static pages that are written in the Hypertext Markup Language (HTML), is not suited for the great body of data that already resides in database management systems (DBMS) for several reasons. Potentially huge data volumes must be managed twice, data updates are difficult to handle, and search facilities are limited. Another limitation is that standard WWW technology does not allow to combine partial results from different servers in a single result page.

The restrictions sketched above become more and more apparent to both software developers and users as they get more experience when developing and using WWW software. Fortunately, several technologies that are supported widely, address these problems.

The tutorial is organized as follows. In Section 2 these technologies, namely World-Wide Web (WWW), the object oriented programming language Java, and the Object Management Group's Common Object Request Broker Architecture CORBA, are introduced. Section 3 provides an overview on different approaches to access databases on the Web. Additionally, criteria to compare these different approaches are proposed. Several case studies from projects at FZI, in which these approaches have been used, are presented in Section 4. A short summary concludes the tutorial (Section 5).

2 Base Technologies

2.1 The World-Wide Web

The World-Wide Web provides transparent access to distributed documents. It follows a strict client-/server architecture. The stateless Hypertext Transfer Protocol (HTTP) is used for the communication between the client (browser) and the WWW server. The most common format to present information on the Web is the HyperText Markup Language (HTML), a standardized language for creating formated hypertext documents, which can even include executable content.

These documents are addressed by using Uniform Resource Locators (URLs). Additionally to such documents, i.e., static HTML pages, Web servers can as well provide transparent access to other information sources such as DBMSs. For this purpose the Web server communicates with application programs via the standardized Common Gateway Interface (CGI) [11]. This mechanism allows clients to submit queries that in turn are processed at the server site by the Web server and the database server.

2.2 Java

Java [2] is an object-oriented programming language that has been especially designed to be used on networks of heterogeneous computers (platform independence). Java can be used both stand-alone and in conjunction with the World-Wide Web. Small Java programs, so-called Applets, are transferred as byte code from a Web server to a browser using HTTP. These Applets are executed by a virtual machine that runs in the browser. Just-in-time compilers improve the performance. The execution model ensures a fairly good degree of security even for Applets that are downloaded across the Internet.

2.3 Common Object Request Broker Architecture

The *Common Object Request Broker Architecture* (CORBA) is a standard for open distributed systems that is defined by the Object Management Group (OMG) [13]. It defines ways for objects and clients to interact within a distributed environment [16, 12].

The main features of the CORBA specification are a core object model, localization transparency (clients and server need not be aware of their respective locations, e.g., at different hosts), and programming language independence. These are realized by providing a specific *Interface Definition Language* (IDL) as well as a Dynamic Invocation Interface (DII) for objects. CORBA's *Internet Inter ORB Protocol* (IIOP) enables object request brokers from different vendors to communicate with each other.

Furthermore, CORBA *services* and CORBA *facilities* are specified. CORBA services are essential services concerning security, transaction handling, naming, etc. If a system provides these services, they must be CORBA-compliant. CORBA facilities are higher-level services such as desktop management and time operations that are not mandatory.

SIGMOD '97 AZ,USA
© 1997 ACM 0-89791-911-4/97/0005...$3.50

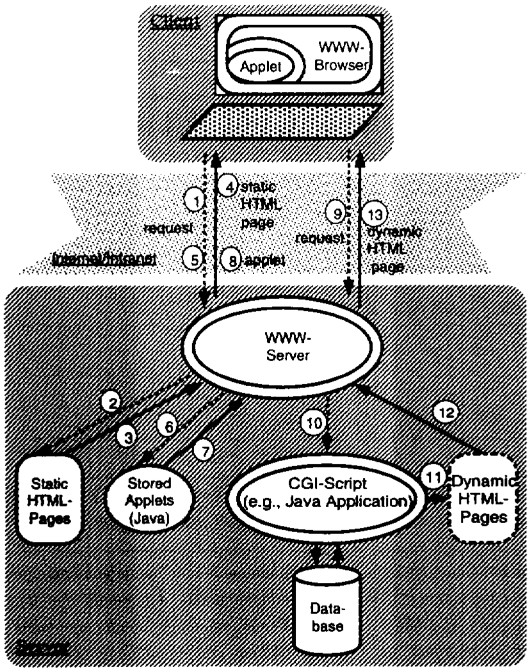

Figure 1: Using static HTML Pages, Java Applets, and CGI Scripts

2.4 Converging technologies

In order to combine the benefits of Java with its strength at the client-site and CORBA with its strength at the server-site, several vendors currently integrate CORBA and Java. The IIOP can be used for the communication between Java Applets that act as CORBA clients and CORBA objects. An example of this integration is VisiBroker for Java [18], an object request broker that connects CORBA and WWW. VisiBroker for Java is based on the CORBA implementation VisiBroker for C++ [17] and written in Java, objects can be implemented in Java or C++ [19].

3 Databases on the Web

3.1 Approaches

Several techniques and numerous tools for accessing databases from the Web are available. With respect to the underlying principles they can be organized into three groups. Most approaches rely on standard Web browsers, HTML, HTTP and CGI. They mainly extend the server site and hence are called *server site approaches*. Motivated by the available computing power at the client site, most recently *executable content approaches* became available. The previous two approaches focus on accessing single databases. Additional *middleware layers* allow to combine data from distributed data sources.

In the following sections, we discuss some representative implementations of each of these approaches. Figure 1 illustrates accessing static HTML pages (arcs 1–4), loading Java Applets (arcs 5–8), and the generation of HTML pages using CGI scripts (arcs 9–13).

3.1.1 Server Site Approaches

The oldest and probably most widely used approach to access databases from the Web is based on the Common Gateway Interface (CGI) [14]. CGI is a standard that allows Web servers to access arbitary external applications, so-called CGI scripts. CGI is supported by all Web servers.

A CGI script is executed at the server site. It is started by the Web server upon client request. The CGI script receives the parameters supplied by the user and hidden variables, generates HTML pages dynamically (e.g., on the basis of data retrieved from a database by the CGI script), and delivers them to the Web server. The Web server then transfers the generated HTML pages to the WWW client.

CGI scripts can be implemented in any programming language. However, to ensure portability of the application at the server site, widely supported technologies should be used. One possibility to achieve this is to implement CGI scripts in Java as suggested for Web-CDS [9].

To overcome some of the deficiencies of the CGI approach (e.g., connect/disconnect to/from the database for each database access), numerous CGI tools are available. One of them is Web* [1], which supports HTML templates with embedded scripts. Scripts that are implemented in Tcl (Tool Command Language) allow variables and control structures. Web* formats the HTML parts and results of the executed Tcl scripts according to the HTML template. Extentions of Tcl such as OraTcl follow the same approach.

3.1.2 Executable Content Approaches

Programs written in Java allow for more elaborate interaction with database systems, basically because they allow to overcome the limitations of the stateless HTT protocol. Hence, the combination of Java and databases is of great importance. Among the various approaches, the most important one is Java Database Connectivity (JDBC) [3], the establishing, open standard for accessing relational database systems from Java. JDBC is an API for database access that is supported by several vendors.

Additionally to compiled languages such as Java and Active XControls, there are scripting approaches such as JavaScript. These scripts are interpreted and executed by the browser, hence allowing for more interactivity and avoiding communications across the net, e.g., for consistency checks of inputs.

3.1.3 Middleware Layers

There are two major approaches to make databases available on the Web using CORBA. The first one uses CGI scripts, which act as CORBA clients. These CORBA clients call a CORBA server that in turn accesses a databases system. The second approach uses the IIOP for a direct communication between WWW client acting as a CORBA client and a CORBA server. Additionally to database systems, arbitrary further systems and applications can be intergrated as well using CORBA.

3.2 Comparison of Approaches

To compare the various techniques for accessing databases from the Web, criteria are proposed that are as much as possible independent from specific technological issues. *Support for different database features* includes support for session management, transactions and the range of functionality supported. *General features* include security, performance and user-friendliness. *Software costs* include training, development and maintenance costs. *Openess* addresses the support for heterogeneous environments, portability, and support for multiple vendor products.

4 Case Studies

4.1 Using CGI and HTTP: WWW-UDK

The environmental data catalogue UDK is a standard meta-information system for environmental data for use in public authorities. The original UDK consists of a database and an application program. At FZI, the UDK database has been made available for both the Internet and Intranets using the tools and techniques of the World-Wide Web. WWW-UDK [4, 6, 7] features several query modes for UDK objects and addresses, an environmental thesaurus, on-line access to underlying data, e.g., databases and environmental reports, multilingual query and result forms, and a hypermedia online help system. Currently, three Ministries of the Environment provide their UDK data via WWW-UDK on the Internet. These and future installations can be reached starting at the URL http://www.fzi.de/dbs/applAreas/eis.html.

Currently, instances of WWW-UDK use different relational DBMSs (Oracle, Informix) and several operating systems (SunOS, Solaris, HP-UX, and Digital Unix). The underlying technologies of WWW-UDK are CGI scripts implemented in C with embedded SQL statements.

4.2 Using Java and JDBC: SOIA

The design and implementation of a JDBC driver for Oracle is explained [8]. A prototyp application to access a socio-economic database (SOIA) is based on Java and JDBC. It offers graphical representation of the data. Screen elements such as dynamic selection lists and bar-diagramms are part of a JDBC based class-library that is currently under development.

In contrast to pure HTML, Java requires a substantial effort in developing graphical user interfaces whereas, on the other hand, Java provides comfortable, e.g., multi-window, user interfaces. The number of communications between a WWW browser and the WWW server can be reduced whereas the communication volume is likely to increase, if data compression is not used. With respect to database accesses in WWW environments, in contrast to the HTT protocol, Java allows stateful client server communication, which make it easier to support database transactions across the net that access multiple tables.

4.3 Building a Federation Architecture: WWW-UIS

In a federation architecture, autonomous components are loosely coupled to build an integrated system. Federation architectures extend the idea of federated and multi database systems (FDBMS) [15] that have been proposed to make information from different database management systems available under a common frame, in which each of the participating DBMS still retains its autonomy.

Figure 2 presents an example of a federation architecture. As an extension to FDBMSs, not only databases ((x)DBS), but also geographic information systems (GIS), expert systems (XPS), and further applications take part in the federation. CORBA is used as the integrating middleware layer. Furthermore, WWW-UIS incorporates event detection based on ECA rules [20], as known from active database systems.

Horizontally, we distinguish between user-level and system-level services. System-level services provide basic functionality, such as accessing databases and preparing HTML pages. User-level services basically combine several system-level services into higher level services that are made available to the user. Java Applets communicate with their server site based counterparts using the IIOP.

Vertically, we distinguish between information and data services. Information services are based on so-called meta data, i.e., data about data. They help the user to find relevant data sources, e.g., data services and reports. Data services access different kinds of data. Both information services and data services deliver results that are used as input for the generation of HTML pages.

5 Conclusions and Outlook

In this tutorial, different approaches for accessing databases from the Web were presented. As these approaches differ in several criteria, no general recommendation for a single technique can be given. However, if only limited functionality for accessing a database is required, the classical CGI and HTTP approach seems sufficient. For enhanced portability at the server site, it can be combined with Java and JDBC. In contrast, if more elaborate functionality and more interactivity is required, e.g., for subsequent updates in several database tables that are represented in different forms, using Java at the client site (i.e., browser) and at the server site is a promising approach.

Most recently, a convergence of open Internet technologies such as WWW, Java, and CORBA addresses many of the problems of technical heterogeneity. These technologies provide a good basis to address the challenges of semantic heterogeneity, e.g., by using mediation techniques [21] to integrate different thesauri as part of the information services (see Figure 2) [5].

Acknowledgments

I would like to thank all members of the DBS group at FZI (especially Arne Koschel, Gergely Lukacs, Ralf Nikolai, and Claudia Rolker), Prof. Dr. P.C. Lockemann, and all students for contributing their ideas and effort to the projects and for the stimulating discussions.

References

[1] Concurrent Engineering Research Center(CERC). Web* home page, 1996. http://webstar.cerc.wvu.edu/.

[2] J. Gosling, B. Joy, and G. Steele. *The Java Language Specification.* Addison-Wesley, 1996.

[3] G. Hamilton and R. Cattell. JDBC: A Java SQL API. Technical report, Sun Microsystems Inc., 2550 Garcia Avenue, Mountain View, CA 94043, U.S.A., May 7 1996. URL: ftp://splash.javasoft.com/pub/jdbc.ps.

[4] R. Kramer and T. Quellenberg. Global Access to Environmental Information. In R. Denzer, D. Russel, and G. Schimak, editors, *Environmental Software Systems; Proceedings of the International Symposium on Environmental Software Systems, 1995*, International Federation for Information Processing (IFIP), pages 209 - 218, London, 1996. Chapman and Hall.

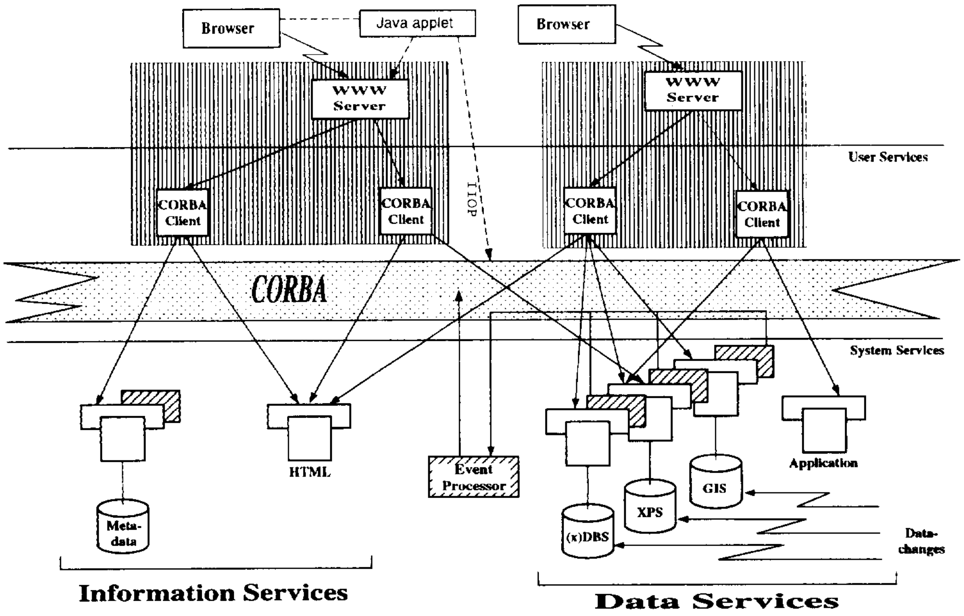

Figure 2: Federation Architecture incorporating the World-Wide Web, CORBA, and Java

[5] Ralf Kramer, Ralf Nikolai, and Corianna Habeck. Thesaurus Federations: Loosely Integrated Thesauri for Document Retrieval in Networks based on Internet Technologies. *Journal on Digital Library*, 1997. to appear.

[6] Ralf Kramer, Ralf Nikolai, Andree Keitel, Rudolf Legat, and Konrad Zirm. Enhancing the Environmental Data Catalogue UDK for the World Wide Web. In Lessing and Lipeck [10], pages 59–68.

[7] Ralf Kramer, Ralf Nikolai, Arne Koschel, Claudia Rolker, Peter Lockemann, Andree Keitel, Rudolf Legat, and Konrad Zirm. WWW-UDK: A Web-based Environmental Metainformation System. *ACM SIGMOD Record*, March 1997. to appear.

[8] Ralf Kramer, Ralf Nikolai, and Claudia Rolker. Java-Evaluierung für das UIS Baden-Württemberg – Phase 1. FZI-Bericht 5/96, Forschungszentrum Informatik (FZI), Karlsruhe, Germany, May 1996.

[9] Ralf Kramer, Ralf Nikolai, and Claudia Rolker. World-Wide Web Access to CDS: Software Design and Demonstration of First Prototype. In *Proc. 3rd Workshop on Catalogue of Data Sources (CDS) and Thesaurus*, pages 81-91, Copenhagen, Denmark, September 1996. European Environment Agency, European Topic Center on Catalgue of Data Sources.

[10] H. Lessing and U.W. Lipeck, editors. *Informatik für den Umweltschutz; 10. Symposium, Hannover 1996*, number 10 in Umwelt-Informatik Aktuell, Marburg, 1996. Metropolis.

[11] Rob McCool. Common gateway interface overview. Technical report, National Center for Supercomputiong Applications (NCSA), http://hoohoo.ncsa.uiuc.edu/-cgi/overview.html, 1994.

[12] T. J. Mowbray and R. Zahavi. *The Essential CORBA*. John Wiley & Sons, Inc., New York, U.S.A, 1995.

[13] Object Management Group. The Common Object Request Broker: Architecture and Specification, Version 2.0. OMG Document, Object Management Group, Inc. (OMG), July 1995.

[14] Jeff Rowe. *Building Internet Database Servers with CGI*. New Riders Publishing, 1996.

[15] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogenous, and autonomous databases. *ACM Computing Surveys*, 22(3):183-236, September 1990.

[16] J. Sigel. *CORBA Fundamentals and Programming*. John Wiley & Sons, Inc., New York, USA, 1996.

[17] Visigenic. VisiBroker for C++. `http://www.visigenic.com/ prod/vbcpd.html`.

[18] Visigenic. VisiBroker for Java. `http://www.visigenic.com/ prod/vbjpd.html`.

[19] A. Vogel. WWW and Java - Threat or Challenge for CORBA? *Middleware Spectra, Spectrum Reports, Winchester, UK*, May 1996. `http://www.dstc.edu.au/AU/staff/andreas-vogel/papers/mws96/paper.html`.

[20] Günter von Bültzingsloewen, Arne Koschel, and Ralf Kramer. Active Information Delivery in a CORBA-based Distributed Information System. In Karl Aberer and Abdelsalam Helal, editors, *Proc. First IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, pages 218–227, Brussels, Belgium, June 1996. IFCIS, IEEE Computer Society Press, Los Alamitos, California.

[21] Gio Wiederhold. Glossary: Intelligent Integration of Information. Technical report, Intelligent Integration of Information (I3), `http://www-db.stanford.edu/pub/gio/1994/vocabulary.html`, 1995.