Paper #3901

Data Warehousing and the Web:

Extending the Data Warehouse
Across the Enterprise and Beyond

George Ferguson

Hewlett-Packard
19490 Homestead Road
Cupertino, CA 90014
U.S.A.

The Explosion of Data Warehousing

Data warehousing has exploded in the market place over the past several years. In February 1993, industry analyst Metagroup found that 5 percent of all surveyed companies (roughly corresponding to the Fortune 1000) had active data warehousing projects. Two years later, 95 percent of the surveyed companies had active warehousing projects!

Companies have jumped into data warehousing in order to establish information-based competitive advantage (or avoid competitive disadvantage) in the global-competitive market place of the 1990's. A recently completed IDC study found that this hope of harvesting the value of information is quite realistic with the mean 3 year return on investment (ROI) of data warehousing projects being 401 percent! (Steve Graham, *The Foundations of Wisdom: A Study of the Financial Impact of Data Warehousing*, (Toronto: International Data Corporation, 1996), p.5).

As data warehouses continue to grow in prevalence and user base, new trends in data warehousing are beginning to appear. One of the most interesting of these is web-browser access to data warehouses.

This paper will examine the following:

the reasons driving companies and organizations to provide web-access to their data warehouses

the limitations of current web-warehouse solutions

the means of building a production web-warehouse environment, including ease-of-use, management, and security

the directions that web-warehouse access will be taking over the next several years.

Reasons for Web Access to Data Warehouses

A number of organization and business needs are driving web-browser access to data warehouses. Among these needs are the following:

1) Need to expand warehouse access to much larger groups of knowledge workers

A data warehouse is intended to empower knowledge workers (i.e. personnel who make product, service, or organizational recommendations and decisions) to help them in making faster, better recommendations and decisions. Part of the value of a data warehouse is dependent on the degree to which the information within the warehouse is made available to the organization's knowledge workers. In fact, the IDC study previously cited found that lack of wide-spread usage was one of the reasons behind data warehouses with negative ROI.

Traditional decision support systems in the 1980's were typically used by only 2 percent of all corporate knowledge workers--primarily the financial analysts, marketing analysts, and a few executives (using pre-canned executive information systems).

Today in companies with deployed data warehouses, the percentage of knowledge workers today actually using the warehouse has typically risen to around 10 to 25 percent. This represents a huge percentage of knowledge workers who are still largely untouched by the benefits of direct warehouse-information access. Forward-looking companies understand this and are actively making plans to roll their data warehouse out to much higher percentages of
their knowledge workers.

2) Need to reduce client software installation and maintenance costs

One of the limitations to more widely deploying data warehousing access is the very high cost of installing and maintaining complex client/server software. The cost of installing and (as versions change) updating client software is very expensive. Companies would like to eliminate or minimize these costs.

3) Need to provide hardware/operating system independence

Many, if not most, IT departments would like to be able to deploy desktop software that is independent of client hardware platform and operating system. In spite of the dominance of Windows/Intel platforms, the Macintosh continues to have die-hard supporters in core groups such as graphics departments and some executives. Likewise, the UNIX workstations is definitely still the desktop of choice in most engineering departments. Hardware/operating system independence would give IT departments one less headache.

4) Need to selectively open data warehouses to users outside the corporation

A very interesting new trend is the desire of companies to open up their data warehouse(s) to selected users outside of their enterprise. This trend is primarily being driven by the desire for companies to share certain information with business partners, suppliers, distributors, and major customers.

For example, MasterCard maintains a massive data warehouse of credit-card transactions. MasterCard is now opening up this warehouse to some of its top merchant banks (banks of the caliber of Chase Manhattan, Citibank, etc.) so that these banks can view the credit transactions of consumers using their respective bank's card and also some overall summary information.

Similarly, large telecommunications companies are now beginning to provide selective access to their data warehouses to allow their major corporate customers to do analysis on their corporations' vast number of phone calls. Manufacturing companies are in like fashion interested in selectively opening their data warehouses to distributors and OEM's. Even governmental organizations and research/educational institutes are interested in opening data warehouses to people outside of their organizations.

All of these companies and organizations see major benefit in selectively opening their warehouses to outside users. Of course in doing so, they want to absolutely avoid having to manage client desktop software outside of their organization.

Each of these four needs can be met through web-browser access to the data warehouse. To understand how web browsers and servers can meet these needs and to set a foundation for understanding how web technology will evolve in the future, it would helpful to do a quick basic review of what is the *Web* and how it works.

World-Wide Web Basics

Web browsers applications are available for nearly every desktop platform and operating system. When a user wants to request information from the web browser, the user specifies to the browser a Universal Resource Locator (URL). This is a special address that is accessible anywhere within a company's *intranet* (a company's full set of internal computer networks) or the global *internet* (the world-wide set of interconnecting networks spanning the globe). For example, Hewlett-Packard's URL is http:\\www.hp.com.

After the user specifies the URL, the browser requests via Hyper-Text Transfer Protocol (HTTP) that the web server display the page of information specified by the URL on the user's screen (within the browser window). The page of information is encoded in Hyper-Text Markup Language (HTML)--a platform and operating-system independent format that describes text, lines, images, etc. The downloaded HTML page is finally displayed on the user's screen (no matter what platform or operating system is in use) since all web browser applications have the capability to display HTML pages on a screen.

One of the interesting properties of HTML pages is that they may contain spots or fields that when clicked on (with a mouse) invoke new URL's and thus new HTML pages on a variety of web servers. Thus, one HTML page make point to (or have hyper-text links to) several other HTML pages which in turn point to yet other HTML pages, some of which may even point back to the original page. The concept of all of these pages in the world being interconnected and pointing back and forth at each other is what is known as the *World-Wide Web.*

The page-oriented nature of HTML is significant because although it is very important to the current nature of the World-Wide Web and the Internet, it is in fact a very limiting paradigm. Standard HTML does not easily support multiple windows or even simple concepts like tables. User interfaces designed with HTML are much less elaborate than MS Windows, Motif, or Macintosh graphical user interfaces. As discussed later in this paper, major changes are coming on the Internet to overcome these limitations of HTML.

The primary means today of accessing a database (including a data warehouse) from a web browser is through the Common Gateway Interface (CGI). CGI is a totally generalized gateway interface that allows some number of parameters to be passed to some program when an HTML event occurs (typically this would be a user clicking on a button on the HTML page). CGI is a very primitive interface. It maintains no state across calls to it (e.g. for database connection reuse) and, in fact, has no concept of databases. It just knows that it is supposed to invoke some specified program and pass to it the specified parameters. In spite of its limitations, CGI is supported in all web servers and is used by nearly all current web-based database access tools.

When used for accessing a database, CGI will typically pass parameters that are used for forming a SQL statement which will, in turn, be passed by the invoked program to a database driver, such as Sybase OpenClient or Oracle SQL*Net for submittal to a database. On return from the database, the results from the database will be built by the invoked program into a dynamic HTML page for display on the web browser.

Several proprietary API's have been created by web-server vendors as alternatives to CGI to overcome some of CGI's shortcomings. Most notable among these are Netscape's NSAPI and Microsoft's ISAPI. Microsoft's Internet Information Server (IIS), in particular, includes an interface from ISAPI to ODBC--the universal database connectivity API. These new, proprietary API's are not yet widely used, however.

In spite of current limitations, we can now see how web browsers can access databases (including data warehouses) using a totally generalized interface that runs on almost every client platform and operating system. What may not be clear yet is the great advantage that web-based applications have over current client/server applications in terms of minimizing client software installation and maintenance.

When a set of HTML pages and associated (CGI-invokable) web-server applications are installed on a web server, they are available to anyone on the intranet or internet, as the case may be. Thus, once web browser software has been installed on individual desktop systems, any number of web applications can be *installed* by simply loading them on one web server. Within Hewlett-Packard, we can install a web-server application *once* and have it immediately available to over *86,000 users*. Of course, the real benefit is not just in installation but in maintenance and new versions.

Combined with the internet, web-server applications allow a company to effectively deploy an application to anyone and everyone in the world on the internet (assuming they have proper security access). This is of huge benefit in opening up data warehouses to users outside of a company or organization.

Thus the benefit of web environments to data warehousing is that they provide global or organizational warehouse access with minimal installation/maintenance costs and full client independence.

Web-Based Data Access Applications

Understanding now the benefits, it is instructive to see what web-based database access applications currently exist. The following table shows some of the major offerings and their category:

Company	Product	Category
Arbor Software	Essbase Web Gateway	OLAP (MDD)
Business@Web	OpenScape	Visual App. Builder
Information Advantage	Decision Suite (WebOLAP)	OLAP (ROLAP)
Information Discovery	Map/IDIS	Data Mining
IQ Software	IQ/LiveWeb	Production Reporting
Oracle	Discover2000, Express	Visual App. Builder; OLAP (MDD)
Speedware	Autobahn; Media/Expressway	4GL; OLAP (EIS)
Spider Technologies	Spider 2.0	Visual App. Builder

With these tools, users can access all of the major databases used for data warehousing (e.g. Informix, Oracle, Red Brick, Sybase, etc.).

Production Web Warehouse Access Requirements

In spite of the major strides that have been taken very recently in opening up data warehouses to web-browser access, a number of additional web warehouse access issues continue to exist--particularly relating to production environment requirements. Companies implementing web access to their data warehouses should carefully examine alternatives for dealing with these issues.

1) How to manage large numbers of web warehouse users

One of the drivers for web access to a warehouse is serve large numbers of users. The idea of a large number of unmanaged warehouse users should be frightening, however, to most IT departments. Unlike transaction processing transactions which are typically small, quick update transactions, data warehouse transactions are typically large, complex queries. Single queries can easily run for tens of minutes or even hours. Within this large-scale ad hoc environment, it is extremely important to manage the environment through:

the maintenance of usage pattern information to assist the administrator in warehouse tuning through summary population and index generation

automatic aggregate navigation functionality to optimally map user queries to the currently existing summary tables

query blocking mechanisms to immediately reject non-sensical queries that could consume enormous system resources

2) How to serve management needs of both web and non-web clients

In spite of the advantages of web access, companies will typically (and for good reason) need to manage warehouse access for both web and non-web users. For the next couple of years, at least, power users will continue to need the more robust native graphical user interfaces of their desktop platforms and will also need their full suite of integrated analysis tools. More casual users will be more likely to use a simple web interface. IT departments will need to be able to manage both sets of users--preferably from a common management station.

3) How to provide ease-of-use within the limitations of HTML thin clients

Web warehouse users will typically be casual users and require high ease-of-use. As has already been established, however, HTML is a rather limited interface. Furthermore, most ease-of-use capabilities have been implemented on desktop platforms--not on the servers where web applications reside.

Ideally ease-of-use (which is largely based on metadata mappings between the physical database tables and columns and the business views shown to end users) should be implemented using a generalized set of server-based metadata that can enable the variety of web and non-web tools that organizations will be using to meet various organizational needs.

A few alternatives exist for dealing with some of these issues from a single tool basis. Companies using multiple tools or categories of tools should definitely evaluate HP's

Intelligent Warehouse management software, which provides all of the production capabilities just mentioned.

4) How to provide internet access to an internal data warehouse without opening a major security hole

Having already established the need for external access to internal data warehouses, the issue here is how to allow such access without opening a major security hole into the entire corporate intranet. Typically, externally focused web servers are placed outside of a company's corporate firewalls precisely to avoid this security hole. The difficulty is that it also does not make security sense (in most cases) to put an entire data warehouse outside the corporate firewalls.

To deal with this issue, HP provides its VirtualVault technology. VirtualVault is a secure web-server platform that sits on the boundary between the internet and intranet. The only communication between the outside and inside world is a secure CGI gateway that passes the database calls (or parameters for database call generation) from the outside to the inside. All other access to the intranet world is eliminated. Furthermore, the VirtualVault's HP-UX platform has no single central point of control, no superuser, no root, and no rlogin capabilities for accessing remote platforms. VirtualVault provides full B1 military-level security to ensure that even authenticated users have access to no
other resources besides the data warehouse that they are authorized to access.

Web Warehouse Futures

As has been established, HTML is a limited interface for constructing complex graphical interface applications. This is widely recognized in the industry and has given rise to two primary competing technologies to at least assist, if not completely take over the web mantle from HTML.

Java and JDBC

The first of these technologies is Java from JavaSoft. Java is an interpreted language that is based on a constrained version of C++. Netscape, which has an 80 percent share of the web-browser market, has included a Java interpreter in its Netscape Navigator Version 2 web browser. Under market duress, Microsoft is also including a Java interpreter in the Microsoft Internet Explorer 3.0 web browser. Thus Java has the nearly the same platform and operating system independence as does HTML. Similar to HTML hyperlinks, Java applets may be invoked (via a URL) from an HTML page or from within another Java applet.

Java did not originally provide a means of providing database access, but on March 8, 1996, a draft *JDBC* (Java Database Connectivity) specification was published. The JDBC specification was immediately endorsed by most major web and DBMS vendors, with the notable exception of Microsoft. This is ironic since JDBC is actually a close variant of Microsoft's ODBC API. Details on JDBC can be
viewed on the web at http://splash.javasoft.com/jdbc/. JDBC is a standard SQL database access interface that provides Java programmers with a uniform interface to a wide range of relational databases. As described in the specification, JDBC will either directly call native database drivers or invoke the more general ODBC API. The specification is intended to go final on June 8, 1996 and implementations are expected very quickly thereafter.

JDBC is likely to become a de facto standard for the same reason that ODBC became a standard--tools vendors strongly want one interface to use for application development. Individual API's from the DBMS vendors are unlikely to be able to attract major tools support if a generalized API is available.

In spite of the market and technical strengths of JDBC, wide-scale adoption will be slowed by security concerns over Java applets. Although security is intended to be a strength of Java, web and press accounts continue to document serious security concerns. These security concerns will probably be resolved over the next couple of years, but in the mean time companies will be very loth to allow potentially virus-infected floating Java applets on the internet to execute on their desktop platforms.

Java/JDBC applets are likely to appear in secure corporate intranets by early 1997, but general corporate usage on the internet is likely to be much slower.

Microsoft VB Scripts and ActiveX

The main competitor to Java and JDBC is Microsoft. Microsoft is pushing Visual Basic Scripts (VB Scripts) as an easier, machine-independent interpreted-language alternative to Java. (The machine independence will initially be limited, however, to Microsoft and Apple Macintosh platforms.) Microsoft has also licensed Java itself and will pursue it based on market demand. Microsoft will integrate both VB Scripts and Java with the Microsoft component object model (COM) objects through Microsoft ActiveX extensions. As part of this effort ActiveX will provide the API's for database access. As in the case of JDBC, ActiveX database calls will translate at some level to ODBC calls. Thus whichever strategy prevails in the market, ODBC-based infrastructure should be well positioned for the future.

It is clearly too early to determine the winner out of the Netscape/Java/JDBC versus Microsoft/VB Script/ActiveX wars. Most likely, both sets of products will co-exist in the market for some time.

Summary

Web-browser access to data warehouses can provide global or organizational warehouse access with minimal installation/maintenance costs and full client independence. Off-the-shelf web-based data access tools are now appearing and a variety of web-based application development tools also exist to help companies to implement web-browser access to their data warehouses today. Corporate implementations need to be careful, however, to deal with the production issues of query management, security, and ease-of-use. In the future, web-browser user interfaces for data warehouse access will continue to improve as HTML is supplemented or replaced with Java and VB Scripts. Whatever the future holds, however, companies should actively invest in web warehouse access today in order to achieve competitive advantage through opening warehouse access to large groups of their knowledge workers and through selectively opening warehouse access to business partners and major customers.