



Hurwitz BalancedView Report

Data Warehousing and Business Intelligence

The Right Architecture for e-Business Intelligence

Philip Russom

Data Warehousing and Business Intelligence

prussom@hurwitz.com

The Right Architecture for e-Business Intelligence

table of contents

1 Requirements for e-Business Intelligence

- 1 Web-Based e-Business Intelligence Users
- 1 EBI User Interface
- 2 Scalability and Reliability
- 3 Performance and Bandwidth
- 4 EBI and Portals

4 Software Architectures for EBI

- 4 Common Aspects of OLAP Architectures
- 5 Relational Online Analytic Processing (ROLAP)
- 6 Desktop Online Analytic Processing (DOLAP)
- 7 Multidimensional Online Analytic Processing (MOLAP)

8 Hurwitz Group Analysis

- 9 Performance Requirements
- 10 Analysis Requirements
- 10 The Right Architecture for e-Business Intelligence

The Right Architecture for e-Business Intelligence
written by: Philip Russom, Data Warehousing and Business Intelligence
prussom@hurwitz.com

Published by Hurwitz Group, Inc.
111 Speen Street, Framingham, MA 01701 ► Telephone 508 872 3344 ► Fax 508 872 3355
E-mail: info@hurwitz.com ► Web: www.hurwitz.com

July 2000

A Publication of the Hurwitz BalancedView
Data Warehousing & Business Intelligence Knowledge Center
Copyright 2000, Hurwitz Group, Inc.
All rights reserved. No part of this report may be reproduced or stored in a retrieval system
or transmitted in any form or by any means, without prior written permission.

REPDWBI-0700-106

Requirements for e-Business Intelligence

e-Business means conducting business based on complete and up-to-date information, information that is shared widely among employees, customers, and partners. The rise of e-Business has forced changes in both business practices and the technologies that enable them. In the realm of business intelligence, the information flowing through the enterprise and beyond — a flow that is fundamental to e-Business — has accelerated the older trend of “democratizing” information by making it accessible to greater numbers of users and a broader range of users than ever before.

e-Business intelligence — or EBI — has requirements that are driven by the realities of e-Business. To reach the breadth of democratized users, EBI demands that analytic applications and tools be deployed in a web environment, whether Internet, intranet, or extranet. Web environments offer their own challenges to EBI systems, such as limited bandwidth and unpredictable client platforms. Yet an EBI system must overcome these challenges to scale up to high user counts, minimize query and report loads, and present serviceable information on the user’s screen, even when the configuration of the user’s computer is unknown. Because one of the goals of information democratization is self-service, users in a web environment expect 100% availability of an EBI system. All totaled, the requirement list for EBI is daunting, and not easily satisfied by just any software architecture for EBI.

Web-Based e-Business Intelligence Users

Business intelligence (BI) users have, in the past, belonged to user communities that were well defined, easy to understand, and controllable. With EBI, however, today’s users represent greater diversity in terms of their job titles, needs, goals, and physical location. Many e-Businesses open their EBI systems to external users — like customers and partners — which further blurs the definition and controllability of the business intelligence user community.

Hence, when you deliver analytic tools and applications in web environments, you don’t necessarily know or control the user community at the other end. For example, it’s unlikely you can mandate hardware capacity or demand that users install a certain browser type or client application.

EBI User Interface

Given the unpredictability of user platforms, the only way to ensure that information is displayed in a readable and useful way is to avoid special demands of the user platform. In other words, expecting users to install a plug-in is not realistic if you don’t know what kind of browser they use. Downloading a Java applet takes more time than the impatient EBI user will tolerate, and it is subject to the vagaries of the user’s Java virtual machine. Hence, the

safest way to proceed is to provide a pure hypertext markup language (HTML) solution, using HTML that is readable by all browsers and that incorporates an attractive and useful user interface and data presentation.

Exceptions to the “HTML-only rule” involve highly specialized functions that demand processing in the browser or elsewhere on the client computer. For instance, data visualization and other forms of interactive analysis may demand dynamic HTML or Java as enabling technologies. Even so, a very small percentage of the EBI user base needs these specialized functions.

Vendors of BI software sometimes debate “web parity” — the idea that the client/server-based and web-based user interfaces of an online analytical processing (OLAP) tool must have identical functionality and appearance. Client/server BI deployments typically support hundreds of users, whereas web-based EBI deployments are scaling up to thousands of users. The reality is that, when scaling to thousands of users, the deployment reaches broadly across and deeply into an organization, far beyond the limited number of specialized power users for which the client/server version was originally developed. Hence, web parity is nice for power users, but not a requirement for the “democratized” users of large e-Business deployments.

Scalability and Reliability

With EBI, the need for scalability is driven by the great number of users that information democratization and e-Business information flow bring to the system. When an EBI system is open to external users, it can be difficult to predict at which peak times they will access the system in large numbers or when they will run demanding queries. Therefore, an EBI system must scale to high user counts and manage unpredictable queries, even though its administrators cannot anticipate usage patterns.

Scalable Operating Systems

Serious concerns still linger about Windows’ ability to scale. Although Windows 2000 is a good choice as the server platform for departmental analytic applications, flavors of UNIX — because of their proven scalability and reliability — are more appropriate to the high user counts of EBI. Hence, an EBI system’s server should run on popular brands of UNIX, including IBM AIX, HP UX, and Sun Solaris.

Intelligent Load Balancing

One strategy for ensuring scalable e-Business is to implement an EBI system in a multiserver environment. Balancing the load among servers then becomes an issue. Many load-balancing algorithms are simply “round robin” schemes that transfer the load to the server with the

least number of jobs queued up, the assumption being that few jobs means a light load. Such schemes fail to take into consideration the fact that some jobs process quickly, while others take considerable time. For instance, fetching a report page is trivial for server load as compared to an ad hoc query. Intelligent load balancing assesses the estimated time-to-completion of individual jobs before determining which server has the lightest load.

Fail-Over with Self-Healing

Besides scalability, a multiserver environment also can ensure reliability through fail-over. That is, when a server fails, an available server steps in and executes the failed server's jobs. Fail-over is important in e-Business environments, because users expect around-the-clock availability. But fail-over alone is not enough. A multiserver EBI system also must support "self-healing," which restarts the failed server automatically and recovers its work-load without human intervention.

Performance and Bandwidth

Users have high expectations of performance and availability for any application deployed in a web environment. An EBI system must ensure good performance, even when bandwidth is limited.

Streaming Reports

When a user requests a streaming report, the server quickly sends the first page of the report to the browser, and — while the user reads the first page — the server sneaks in and caches other pages. This way, the user perceives a very fast response from the EBI system. Furthermore, relatively few pages are sent over the network, thus preserving bandwidth, which is fundamental to e-Business performance. Page-at-a-time serving is especially key when you have no control over the voracity of the Internet connections on which external user communities rely.

Summary First, Details When Needed

An emerging best practice among designers of analytic applications, dashboards, and portal-user interfaces is to open a report with a summary view, into which users can drill if they need detailed information or want to conduct analysis. For many users, the summary is all they need, so they needn't run queries or access other pages. Therefore, the approach of "summary first" decreases the number and complexity of queries, which in turn preserves bandwidth and reduces server load.

Prebuilt Cubes

It's a well-documented fact that collecting, transforming, and aggregating data for analysis and reporting takes considerable processing time. Doing this on-the-fly usually results in poor

performance, which most users won't tolerate in an e-Business environment. For the sake of performance, an EBI server can maintain prebuilt cubes that are ready for quick access.

EBI and Portals

In e-Business, as many information systems as possible are web-based, so a broad range of employees, customers, and partners can access them. As a way of integrating several web-based applications, databases, and tools, many corporations are now implementing enterprise information portals (EIPs). Long before EIPs, however, BI systems migrated to web environments, evolving into business intelligence portals (BIP) in the process.

A BIP provides personalization, information organization, and collaboration. These are required of any corporate portal, but the BIP provides these specifically for BI content. In many companies, however, a BIP also must feed the EIP with business intelligence content, facilitating the integration via web standards like HTML and extensible markup language (XML). In such cases, the BIP must manage security, personalization, profiling, and data access issues for BI content, although the content is presented through the EIP.

Software Architectures for EBI

An e-Business environment demands that an EBI system maintain high performance despite large user counts, massive data volumes, and complex reports and queries. When an EBI system is used for data analysis, the architecture of its software component for OLAP has great impact on whether the performance goals of e-Business are met, since OLAP is a fundamental enabling technology for EBI.

The concepts behind OLAP have been around for many years now, and several architectures for implementing OLAP have emerged. Software vendors supporting OLAP have, in recent years, been evolving their product architectures to adapt to the Internet. Now these same vendors must further evolve their products to meet the demands of e-Business environments. The ongoing evolution of OLAP architectures has serious ramifications for EBI.

The three major OLAP architectures currently available in the marketplace are reviewed below, and the ensuing Hurwitz Group Analysis assesses their suitability for e-Business.

Common Aspects of OLAP Architectures

Regardless of the approach, all OLAP architectures involve building a multidimensional data structure, wherein dimensions represent business entities like sales regions and products or natural entities such as time periods and geography. It is this data structure — sometimes called a “cube” — that users analyze via an OLAP tool. Two issues regarding the multidimensional data structure distinguish OLAP architectures:

- ▶ **When and how is the multidimensional data structure constructed?** Some OLAP approaches require a data extraction, transformation, move, and load (ETML) process, which typically runs at off-peak usage times to build and update a persistent multidimensional data structure. However, other OLAP approaches access source data directly to build and present multidimensional data on-the-fly as the user performs analysis.
- ▶ **Where does the multidimensional data structure live and how persistent is it?** The multidimensional data structure may reside in a persistent, dedicated multidimensional database, a micro-cube temporarily cached in memory, or a star schema (possibly expanded to a snowflake schema) stored in a relational database.

Relational Online Analytic Processing (ROLAP)

ROLAP-based systems rely on relational databases for data storage, access, and retrieval processes. This alleviates the need to store and calculate data in a proprietary cube or multidimensional database server, but typically requires that a “star schema” or other summary-oriented multidimensional data structure be created and stored in the relational database. A ROLAP engine is an application logic layer that translates user actions into SQL-based queries, manages those queries against source data in relational databases, aggregates data and performs calculations on-the-fly to build summaries, and presents the results to users in a multidimensional format. Representative software vendors supporting ROLAP-based products include Information Advantage, Informix MetaCube, Microstrategy, and NCR Teradata.

Advantages of ROLAP

The ROLAP architecture was designed with broad access to diverse data in mind. In terms of analytic data content, ROLAP straddles the fence, providing analytic access to both summary data (typically in the fact table of a star schema) and the granular data that lies beneath it (typically representing transaction-level detail, customer data, etc.). All the data resides in relational databases (or legacy sources that have a SQL layer), which IT personnel already have in-house and for which they understand the best practices.

For well-understood, recurring queries, IT personnel can build summary tables (and even load these into memory) to ensure quick system responses. An important aspect of any ROLAP system is that the metadata layer also enables wide-ranging ad hoc queries, possibly involving detailed source data. Performance of these is, at best, unpredictable and very poor in worst cases. Ad hoc queries against source systems also may adversely impact the performance of those systems. However, ad hoc query users are a minority within the ROLAP user base, and they tolerate disappointing performance in exchange for the ability to ask unanticipated and unique analytic questions against granular data in very large data sets.

Disadvantages of ROLAP

Designing and maintaining the summary tables for a ROLAP-based solution may entail a hefty ETL process, depending on the implementation. Even when a star schema is designed and deployed in a reasonable time period, the next phase — involving “snowflaking” — is another matter. The fact-tables that constitute the snowflake part of the data model evolve over time, since the need for specialized aggregations are typically discovered as users work with the data. Hence, ongoing, continuous change is a severe issue in maintaining a ROLAP-based solution, and it can delay project completion and lead to budget overruns. This drawn-out evolution of the data model is contrary to the expectations of many users nowadays, because data warehouse projects are often completed in 90 or fewer days.

Reliance on SQL is an oft-touted benefit of ROLAP architectures, under the assumption that SQL is a well-understood and broadly supported technology. However, SQL was designed for one-dimensional, transaction-oriented queries, not for the complex, multipart queries that multidimensional analysis demands. A ROLAP engine must overcome the limitations of standard SQL (as well as database-specific SQL implementations) by generating multipass SQL. This is especially important for cross-tab creation. Multiple queries that run against a relational database require numerous temporary tables in the relational database to store result sets from each pass prior to joining them. Most relational databases support only a few hundred temporary tables, which can severely limit the number of users and the complexity of their queries. This is an issue for e-Business, where user counts range into the hundreds and even thousands.

Desktop Online Analytic Processing (DOLAP)

DOLAP architectures involve the building of proprietary multidimensional cubes. DOLAP cubes may reside in a proprietary multidimensional database or proprietary cache files that are indexed. Many users store DOLAP cubes on client PCs, hence the name “desktop OLAP.” Cubes downloaded to PCs tend to be small — sometimes called “microcubes” — because they are the result sets from a multidimensional query. Claiming that DOLAP is an autonomous OLAP architecture is somewhat problematic, because some DOLAP systems support midtier servers that host DOLAP cubes and others enable access to cubes on multidimensional OLAP (MOLAP) servers. Representative software vendors supporting DOLAP-based products include Brio Technologies, Business Objects, and Corvu.

Advantages of DOLAP

The point — and primary advantage — of the DOLAP architecture is to collect, aggregate, and calculate data in advance of the analytic act, so that a prebuilt cube is ready to give good query performance to analytic users. Although some implementations of DOLAP systems require downloading a microcube to a PC, the PC-based cube yields snappy responses to queries and

enables off-line analysis. Some software vendors of DOLAP systems are migrating their products to a server-based architecture to enhance performance, scalability, and integration with e-Business environments.

In general, cubes contain summarized data, organized in a fixed structure of dimensions. This is ideal for well-understood, recurring analytic questions and reporting, which constitute the bulk of most analytic user bases. However, the ability to drill-through into detailed data is limited in some DOLAP architectures and may require integration with a separate query tool. The ETML process tends to be relatively simple in a DOLAP architecture, which can enable quick deployment of analytic solutions.

Disadvantages of DOLAP

In some DOLAP systems, dimension members must be held in memory to load the cube and service-query requests. For summary analysis in most systems, this is practical only if the number of dimensions is limited and the dimensions do not contain many members. This limitation of the DOLAP architecture can be a challenge to Internet-based commerce and e-Business applications of any size, because they often involve large dimensions with members representing thousands of products and thousands (sometimes millions) of customers.

A DOLAP session usually starts with a query, which means there is always a load on a database and on the BI server. The query from a DOLAP tool returns a cube to the desktop computer, and, although the cube is relatively small, its delivery over a network is detrimental to bandwidth, that precious commodity of e-Business. DOLAP is acceptable in most client/server and intranet deployments, but, when the number of users rises to extranet and Internet levels, it becomes untenable.

Multidimensional Online Analytic Processing (MOLAP)

Like DOLAP, the MOLAP architecture relies on prebuilt, proprietary multidimensional cubes. Unlike DOLAP, however, MOLAP systems host cubes in a dedicated multidimensional database server. Server-based MOLAP systems can support much larger cubes than DOLAP systems can, as well as cubes that are more complex in their number of dimensions, members, categories, and amount of granular (non-summarized) data. Representative software products supporting MOLAP-based products include Applix TM1, Cognos Cube, Hyperion Essbase, and Oracle Express.

Advantages of MOLAP

Again like DOLAP, the primary advantage of the MOLAP architecture is to collect, aggregate, and calculate data in advance of the analytic act, so that a prebuilt cube is ready to give good query performance to analytic users. Advantages over DOLAP include the MOLAP server's ability to relegate processing to the server (required for truly thin clients), host very large and complex cubes (for the granular data that e-Business requires), ensure scalability

(for the high user counts of e-Business), and integrate with the web-based world of e-Business. Furthermore, most MOLAP systems support advanced features like drill-through to operational data (for auditing aggregated sources) and page-at-a-time serving of reports (for reducing network traffic and faster responses).

Disadvantages of MOLAP

The data of a cube represents a bounded world, a limited collection of heavily summarized data within a somewhat inflexible data model. Cubes are designed for asking the same well-understood analytic questions over and over, although optimized for good performance while slicing and dicing multidimensionally. Most cubes are not well suited to data exploration or detailed reporting.

Even so, this is largely a matter of best practices. Although most MOLAP-based business intelligence systems are designed to contain mostly summarized data, this does not necessarily need to be the case. Technological advances with some server-based MOLAP cubes now enable them to include millions of records and thousands of categories, so that considerable amounts of detailed data can coexist with aggregated, summary-level data.

This is an issue for e-Business. When EBI systems are opened to customers, the customers usually ask for reporting against highly detailed data about their transaction history, financial records, or the current state of their outstanding orders. The typical MOLAP cube consists mostly of summary data that is appropriate for executive-level analysis of sales by region or general ledger reporting, but not for reporting against detailed data. Hence, the MOLAP cube of an EBI system must serve the need for summary analysis of its traditional internal audience, as well as the need for heavily detailed reporting, sometimes driven by an external audience that includes customers and partners.

Hurwitz Group Analysis

As noted earlier in this paper, all OLAP architectures involve building a multidimensional data structure. Two issues concerning the multidimensional data structure distinguish OLAP architectures:

- ▶ **Is the multidimensional data structure constructed through a batch-oriented ETL process, real-time ETL, or a combination of the two?**
- ▶ **Is the multidimensional data structure a prebuilt cube in a database/indexed file or a unique cube in memory?**

The way that an OLAP architecture addresses these two issues typically leads to trade-offs in performance vs. analysis requirements. Figure 1 charts these trade-offs.

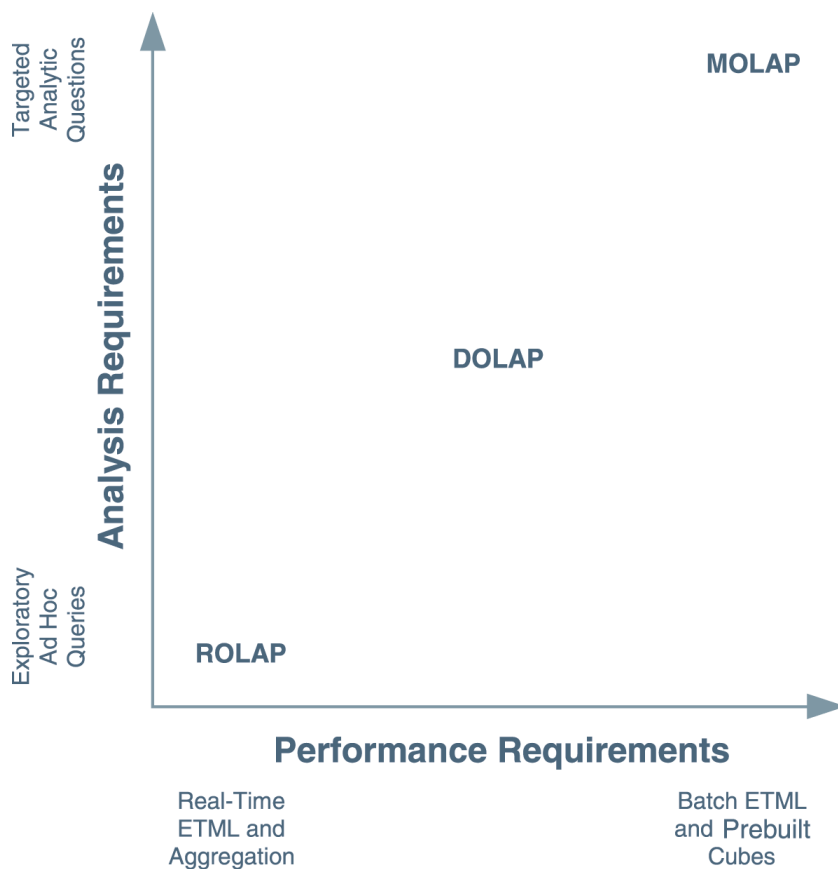


Figure 1. Performance vs. Analysis Requirements in OLAP Architectures

Performance Requirements

(See the horizontal axis in Figure 1.) A prebuilt, persistent multidimensional database (as found in a MOLAP architecture) offers excellent response for analytic queries, enabling users to perform analysis at the speed of their thought processes. Building the dedicated multidimensional database, however, is done through an ETML process that requires expensive tools, specialized IT personnel, and a data movement workflow that demands monitoring and maintenance. Building the multidimensional data structure on-the-fly (as is often the case in a ROLAP architecture) avoids the complexities of the ETML process, but performance suffers to the point that analytic queries run for minutes (even hours!) at a time, derailing the user's train of thought. Some DOLAP architectures ride the fence between these extremes by building cubes ahead of analysis time, but doing so via a simplified and limited ETML process.

Other performance requirements (as discussed in the first section of this paper) focus on scalability for the high user counts of e-Business, which is achieved through multiple EBI servers running on UNIX with intelligent load balancing and failover. EBI performance also is enhanced by bandwidth-preserving features, such as streaming reports and “summary first” best practices for report design.

Analysis Requirements

(See the vertical axis in Figure 1.) Another trade-off concerns the content of the data in the multidimensional structure, and how this affects the scope of analytic questions that the structure can handle. The tendency (although this depends on the individual implementation) is for the persistent multidimensional databases of MOLAP architectures to represent a data model with firm boundaries designed for analytic questions that are well understood and asked on a periodic basis. When ROLAP is applied to building cubes at analysis time, the tendency is to present a broad and open-ended data model, so users can explore data by asking analytic questions that may be unique and unanticipated. Again, DOLAP meets ROLAP and MOLAP half way, by providing cubes (but without the size, richness of data, and scalability of a MOLAP server) and an open data model (but without ROLAP’s support for massive data sets).

Other analysis requirements center on serving the unpredictable user communities of democratized EBI. In particular, user interfaces of pure HTML (when possible) avoid client incompatibility, and an EBI portal integrates BI content with other applications.

The Right Architecture for e-Business Intelligence

In summary, EBI serves hundreds (sometimes thousands) of users who access information that has been precollected and prepackaged for them. Although client/server and Intranet business intelligence deployments often serve a relatively small number of mostly sophisticated users, the wide majority of users in EBI implementations want — and only have the technical depth to cope with — prebuilt reports, analyses, and managed queries. A few EBI users can author reports that draw from a business-friendly metadata layer, whereas a very slim minority of EBI users need open-ended ad hoc query capabilities. Furthermore, EBI users operate with high expectations of performance and reliability in web environments, where bandwidth is fragile and users’ location, hardware, and browser type is unpredictable. Of the three OLAP architectures reviewed here, MOLAP satisfies these EBI requirements the best.

Hurwitz Group, Inc.
111 Speen Street
Framingham, MA 01701

T 508 872 3344
F 508 872 3355

email: info@hurwitz.com
<http://www.hurwitz.com>

Part Number
28898