Two-Tiered Enterprise Decision Support Systems Architecture

Marc Demarest
demarest@hevanet.com

Revision 6.2
November 2001

ABSTRACT

Conventional wisdom suggests that two apparently mutually exclusive architectural alternatives for enterprise-wide decision-support systems (DSSs) exist: data warehousing and data marting. For organizations that want to implement high-quality systems, choosing one over the other represents a major stumbling block. In this article, we suggest that instead of choosing between the two approaches, organizations should blend these conventional approaches in a hybrid multi-tiered strategy. The hybrid approach is suitable for organizations with some or all of these characteristics: multiple, quasi-independent lines of business or business units; a heavy investment in legacy OLTP applications, and a smattering of standalone legacy decision-support applications; and a commitment to commonly shared enterprise business models.

An earlier version of this paper appeared in *DBMS Magazine* in July 1994 (v7 n8 p44) under the title *Building The Data Mart*. This appearance was the first systematic treatment of the concept of data marting and its relationship to data warehousing in the industry. Although the concept of data marting, and the architectural model of combining a single warehouse and multiple marts into an enterprise-wide decision support environment, are now taken for granted by most DSS architects and system builders, at the time the article was published, the concept of marting was extremely controversial.

Warehousing Or Marting?

Data warehousing, in practice, focuses on a single large server or mainframe that provides a consolidation point for enterprise data from diverse production systems. It protects production data sources and gathers data into a single unified data model, but does not necessarily focus on providing end-user access to that data (data warehouse designers often treat end-user access as an afterthought).

Data marting, conversely, deals almost exclusively with servicing a distinct community of knowledge workers. As a model, data marting ignores the practical difficulties of protecting production systems from the impact of extraction, focusing instead on the knowledge worker's need for information from diverse production systems consolidated into a model that reflects the knowledge worker's understanding of the business.

By creating a multitiered decision-support model that blends warehousing and marting, organizations can achieve the long-term benefits promised by data warehousing without compromising the immediate business requirements of data-hungry knowledge workers. This hybrid approach can lower the operating costs associated with legacy systems used for decision support, and, when feasible, enable the removal of those legacy systems from the IS environment. The multitiered architecture can also deliver new value to DSS end users through the judicious application of client/server and GUI technologies.

Finally, the hybrid model can enhance competitive advantage in the marketplace through a Unified Data Architecture (UDA). The UDA lets business decision makers monitor their pieces of the business at the appropriate level of detail, model the cross-organizational business processes that their work affects, and predict the future impact of business decisions on other areas of the enterprise.

Data Warehousing Overview

In the late 1970s, it became apparent that mainframe-based production systems could not support enterprise-wide decision support. These systems fragmented fundamental business "objects," such as customers and markets, into transaction-level detail data spread across many production databases, and they could not sustain the performance levels required by mission-critical applications while simultaneously servicing knowledge workers' complex queries. In addition, the production systems captured only a small part of the data required for business decision making, and they could not support the semi-structured, collaborative nature of decision making in the modern knowledge-based firm.

In response to these shortcomings, decision-support theorists abandoned the idea that decision making was strictly an "executive" function to be supported by host-based executive information systems (EISs). Instead, they began to discuss an alternative strategy for large-scale decision support known as data warehousing, a term coined by W.H. Inmon and popularized by IBM as "information warehousing" and which has since become known as data warehousing

The basic data warehousing architecture interposes between end-user desktops and production data sources a warehouse that we usually think of as a single, large system maintaining an approximation of an enterprise data model (EDM). Batch processes populate the warehouse through off-peak extraction from production data sources, and, in theory, the warehouse responds to end-user requests for information passed to it through conventional client/server middleware.

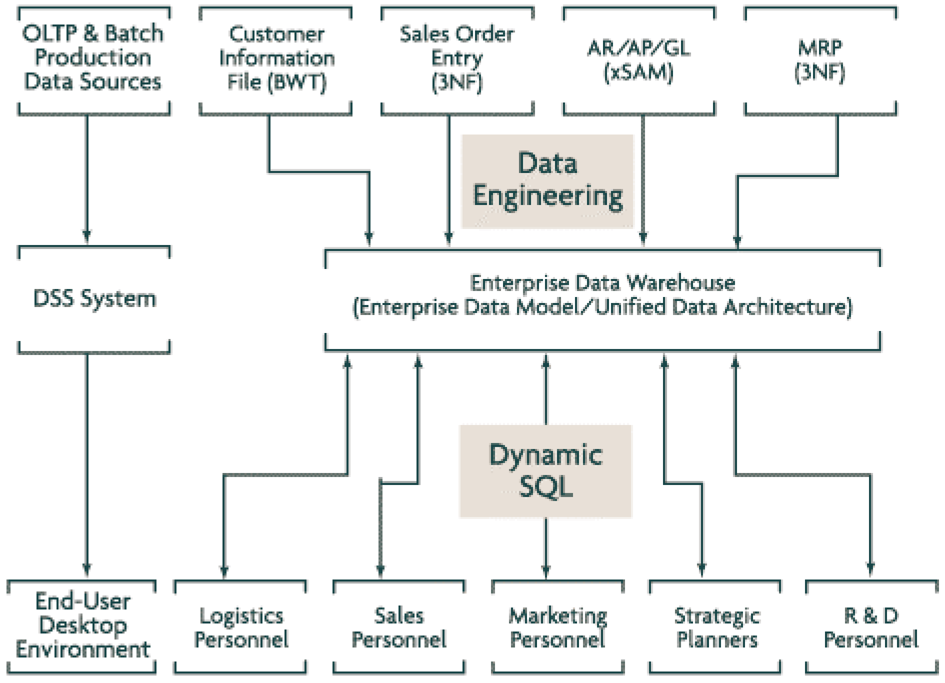
OLTP & Batch Production Data Sources
Customer Information File (BWT)
Sales Order Entry (3NF)
AR/AP/GL (xSAM)
MRP (3NF)
Data Engineering
DSS System
Enterprise Data Warehouse (Enterprise Data Model/Unified Data Architecture)
Dynamic SQL
End-User Desktop Environment
Logistics Personnel
Sales Personnel
Marketing Personnel
Strategic Planners
R & D Personnel

Figure 1 -- Basic Data Warehousing Model

Data warehousing as an architectural model has four fundamental goals:

1. To protect production systems from query drain by moving query processing onto a separate system dedicated to that task, and extracting all the relevant information from each production data source at predictable times when off-peak usage patterns prevail.
2. To provide a traditional, highly manageable data center environment for DSS using tools and practices comparable to those used in data center OLTP.
3. To build a UDA or EDM in the warehouse, so that data from disparate production systems can be related to other data from different production systems in a logical, unified fashion. This would align processes across the organization using a common vocabulary (for example, "customer" would mean the same thing throughout the enterprise), which would simplify the modeling of complex future business behavior and performance based on historical data.
4. To separate data management and query processing issues from end-user access issues so that they can be treated as distinct problems.

Ultimately, the data warehouse can serve as the basis for complex, forward-looking business modeling activities in which strategic planners and business managers use neural networks and other artificial intelligence technologies to surf historical data and predict future business trends.

The data warehouse model has merit, but falls short as a comprehensive decision-support solution.

Let's review its four goals.

1. The first goal of data warehousing (protecting production systems) is a worthy one, and necessary for maintaining adequate levels of performance and end-user response times.
2. The second goal (improving data manageability) also has merit, because DSS data represents a strategic asset of the firm. DSS data is not just a copy of data held elsewhere in the environment; rather, it is a unique, enriched data set.
3. The third goal (creating an EDM) is an honorable but expensive proposition, in terms of both technology and time, and is flawed for nontechnical reasons. Autonomous business units with their own IT functions may refuse to deliver data or metadata to data architects. Legal requirements for different countries or regions may make it impossible to define a standard set of attributes for an entity. Also, affordable tools may not exist to manage data that resides in various proprietary and open system data stores. Finally, any EDM will require constant revision (if not complete renovation) on an annual basis because of rapidly changing business conditions. The cost of this constant management activity may be too high for a firm to bear.
4. The fourth goal (separating functions) also represents a shortcoming of data warehousing. By focusing on data modeling and management and not on direct end-user data access, data warehousing in practice requires expensive paper-based reporting process that SQL and 4GL programmers have to service. Also, data warehousing has historically encouraged the creation of nonportable reporting applications based on proprietary technologies such as 3270 terminals and CICS that today do not represent the leading edge of technology. More important, these technologies do not reflect what business teams use today, such as personal computers, GUIs, and personal and LAN-based workgroup productivity tools.

Data warehouses did not anticipate the dynamism of late twentieth-century markets or the structural realignments required in companies to service those markets. They relied on a static notion of the market and the company, and fixed that notion in a data model. And because data warehouses demanded an impractical degree of completeness, they failed to deliver working data models in a reasonable amount of time.

Ultimately, data warehousing ignored the personal computer and the knowledge worker. As organizations gave more user communities access to the data warehouse, significant warehouse maintenance problems came to light. First, the number of tables increased dramatically (in what we call schema explosion) as each constituent community added new views, summary tables, aggregations, and precalculations, to the transaction-level detail. As a result of this increase in tables, the legibility -- the knowledge workers' ability to navigate the warehouse efficiently -- suffered. In my experience, when the warehouse contains more than approximately 20 tables, the average knowledge worker's ability to navigate the warehouse becomes impaired.

Data Marting Overview

The failure of data warehousing to address the knowledge worker's culture and the practical technical difficulties associated with EDM development and warehouse maintenance prompted Forrester Research in 1991 to declare that data warehousing was dead. It had been replaced by what Forrester called data marting.

Data marts are user community-specific data stores that focus on DSS end-user requirements. In effect, data marting attempts to solve the enterprise DSS problem by presenting only the data that an end-user constituency requires in a form close to the constituency's business model.

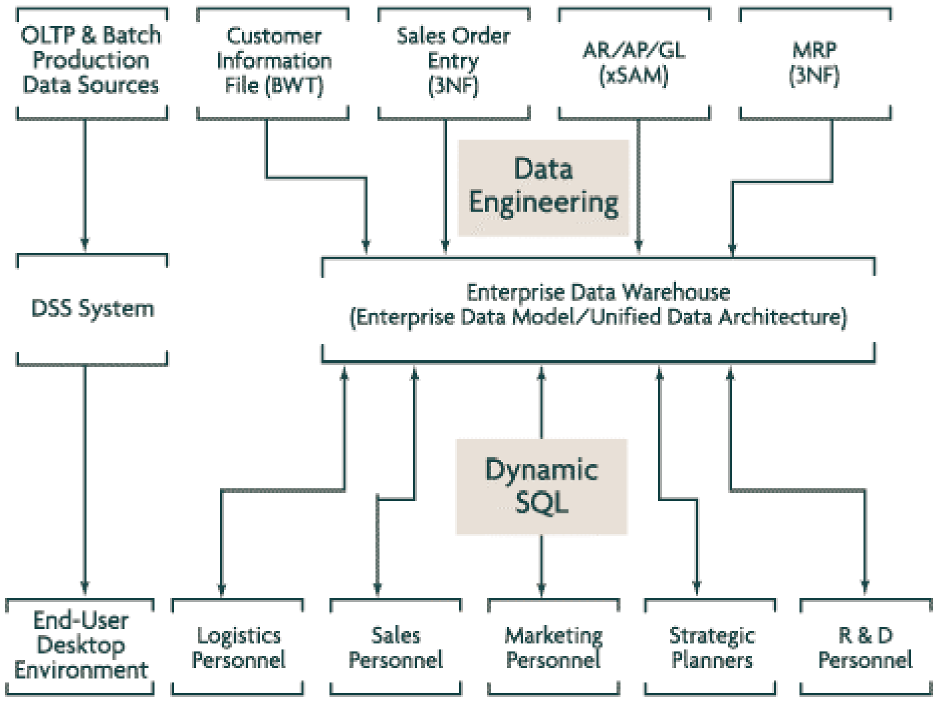
OLTP & Batch Production Data Sources
Customer Information File (BWT)
Sales Order Entry (3NF)
AR/AP/GL (xSAM)
MRP (3NF)
Data Engineering
DSS System
Enterprise Data Warehouse (Enterprise Data Model/Unified Data Architecture)
Dynamic SQL
End-User Desktop Environment
Logistics Personnel
Sales Personnel
Marketing Personnel
Strategic Planners
R & D Personnel

Figure 2 -- Basic Data Marting Model

Data marting has advantages over data warehousing. For example, the data mart focuses on data legibility: The business team sees only the data it needs, in a form that matches its collective understanding of the business. The data mart makes full use of today's LAN-based client/server technologies, integrates with the knowledge workers' toolset, and rides the price-performance curve of those technologies. A data mart provides a homogeneous population of knowledge workers with similar business models, business vocabularies, and responsibilities.

The data marting model supports individual knowledge worker communities quite well. However, it also has disadvantages when seized by LAN independent software vendors (ISVs) as the basis for bidding for corporate IS attention, or by knowledge workers as a pretext for circumventing corporate or divisional IS data architecture efforts by building unauthorized data marts in departments and work-groups outside IS control.

Data marting also overestimates the capability and performance of LAN-based data-management tools. For example, flat-file DBMSs and small-scale RDBMSs do not have the processing power or the facilities to support a high-performance query environment, nor can you connect them to production data sources in a way that allows efficient and timely large-scale extracts. Data marting also neglects -- as do most LAN-based models -- the technical limitations of mainframe-based production systems in providing extract processing time and power. It also neglects the secondary systems and network management morass that is created when many data marts request substantial data extractions from multiple production systems.

Data marting addresses the decision-support needs of only small companies with few knowledge workers, single markets, and simple product lines. On its own, data marting cannot meet the needs of international enterprises with many distinct knowledge edge worker communities, many products and markets, and constant reorganization in response to market conditions.

Perhaps most ironically, data marting neglects the real organizational boundaries drawn in large companies between data management and data movement. While divisional IS organizations can build data marts for their knowledge workers, corporate IS data architects and DBAs may not be willing to populate those marts with corporate data.

Blending the Models

The basis of good enterprise DSS architectures is the concept of information-on-demand: giving knowledge workers the data they need, when they need it, where they need it. For the knowledge worker, data is a raw material that IS organizations are expected to supply as needed. The concept of an "information warehouse" is misleading and inappropriate because information is created dynamically by teams of knowledge workers. It is not stored in a musty repository, among the tuples, waiting to be discovered.

Knowledge workers are the customers of DSS architectures and implementations; data is the raw material they require. Systems architects must deliver the raw material and, to some extent, support the work processes through which knowledge workers transform that raw material into useful information.

Let's consider common distribution schemes as an example. Almost every product we buy as consumers comes to us through a multitiered distribution mechanism. Our local grocer stocks a store based on a detailed understanding of local demographics and buying behavior. The products that fill the shelves of the local grocery come shrink-wrapped on pallets and packed in boxes from a centrally-located warehouse. The warehouse, in turn, is fed by many consumer packaged-goods manufacturers such as candy makers, cereal companies, and bakeries.

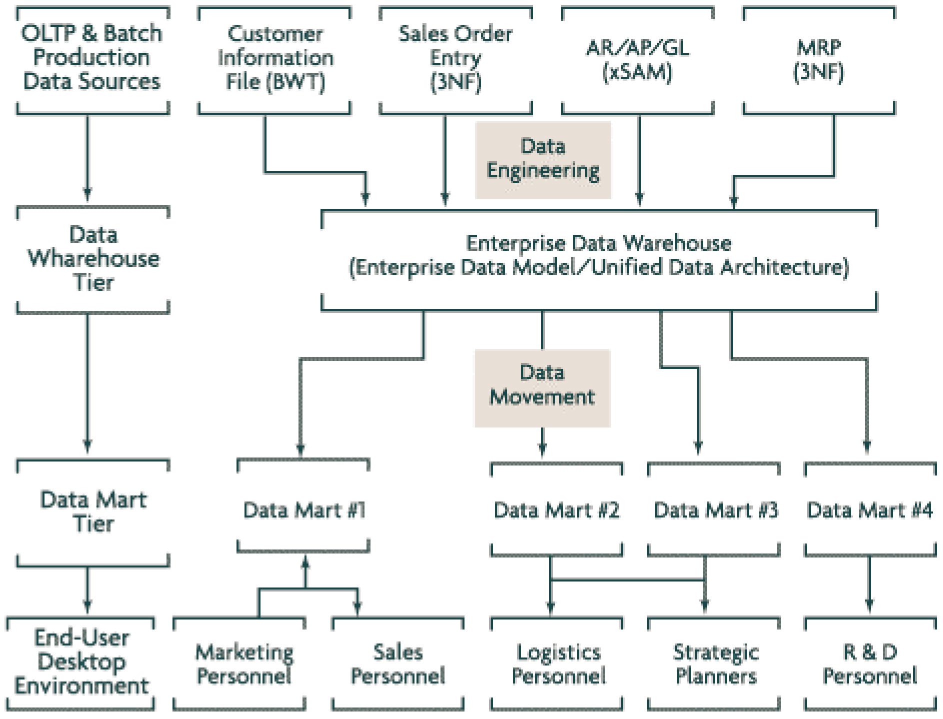

Figure 3 -- Demarest's Hybrid Warehouse/Marting Model

Manufacturers, wholesalers, distributors, and retailers have refined this distribution model for more than 150 years; yet, after two generations of decision-support architectures, information technologists are just now discovering that information is a product, built by knowledge workers from data that is warehoused on the basis of economy and ease of distribution, and retailed based on local need. By understanding this simple idea, you can effectively join data warehousing and data marting into a hybrid enterprise DSS model that offers most of the benefits and avoids most of the limitations of both approaches.

The IS organization must treat the individual business team -- the owner of a product, process, market, or customer -- as the primary customer of the enterprise's information. Only then can IS deliver business information (and the tools to analyze that information) in a supply chain that works for every business team. The IS organization must recognize that each business team requires an appropriately detailed vertical view of the historical data for the product, process, market, or customer for which it is responsible. Each business team requires an aggregated, horizontal view of the whole enterprise, including vertical views of the business teams that constitute its internal suppliers, collaborators, and customers. The team also requires full and seamless integration of the vertical and horizontal views of the enterprise data with the business team's group productivity software and GUI workstations.

In the delivery of decision-support data to business teams, we've identified four distinct processes:

1. Extract all data relevant to the business decision-making processes of groups of knowledge workers from the specific production (OLTP) systems responsible for capturing that information. The extraction process includes extracting copies of the relevant data from the OLTP sources, scrubbing that data to remove anomalies, inconsistencies, and unnecessary information, and enriching the data by translating cryptic numeric codes and acronyms into easy-to-understand textual (and in some cases graphical) information.
2. Store the resulting data sets in one location: the data warehouse. Because the sources of decision-support data are diverse and sometimes unconnected, the data sets created by the extraction process are stored in an EDM. This is a data schema that represents a simplified, but accurate, picture of the major business processes such as the manufacturing/product processes and marketing/sales processes. This model represents only a best-case approximation of the actual enterprise.
3. Create a unique cut or series of cuts of the data warehouse for each knowledge worker community. These are the "data marts," each of which fulfills a different user constituency's business need. For example, customer-support representatives, product marketing managers, and senior managers represent three distinct knowledge markets into which IS must sell different knowledge products. Customer-support reps may think of "customers" as individual names within corporate entities, and measure time in days. Marketing managers may think of customers as corporate entities within distinct market segments, and measure time in months, quarters, and years. Senior managers may think of customers strictly as corporate entities, and measure time in quarters and years. Different knowledge worker communities must view key dimensions of the enterprise at different levels of detail. The detail data in the warehouse must be

rolled up to a higher level, or aggregated, to match each community's understanding of the business.

4. Supply the decision-support tools appropriate to the knowledge workers' style of computing. Returning to our previous example, customer-support representatives' model of data might be electronic and paper forms. Their computer applications may blend decision support and OLTP during a single customer call. In contrast, product managers think of data in terms of electronic spreadsheets, and senior managers think of data in terms of reports, briefing books, or presentations. The skill and comfort levels among knowledge workers also vary. For instance, a customer-service representative may be quite comfortable operating a personal computer, while senior managers may exhibit discomfort or lack interest in using them. This means that you must tailor the decision-support toolset to each information segment that IS serves.

These processes comprise an information manufacturing process and operational model in which corporate IS creates and manages a corporate data warehouse for the divisional and departmental IS organizations. These outlying organizations then tailor the information storage and access for their markets by building data marts and creating both host-based and client/server decision-support toolsets, according to local architectural practice.

Combining data warehousing and data marting, and factoring in the reengineering of production data as well as the complexities of client/server data access and analysis, results in the enterprise DSS model illustrated below.

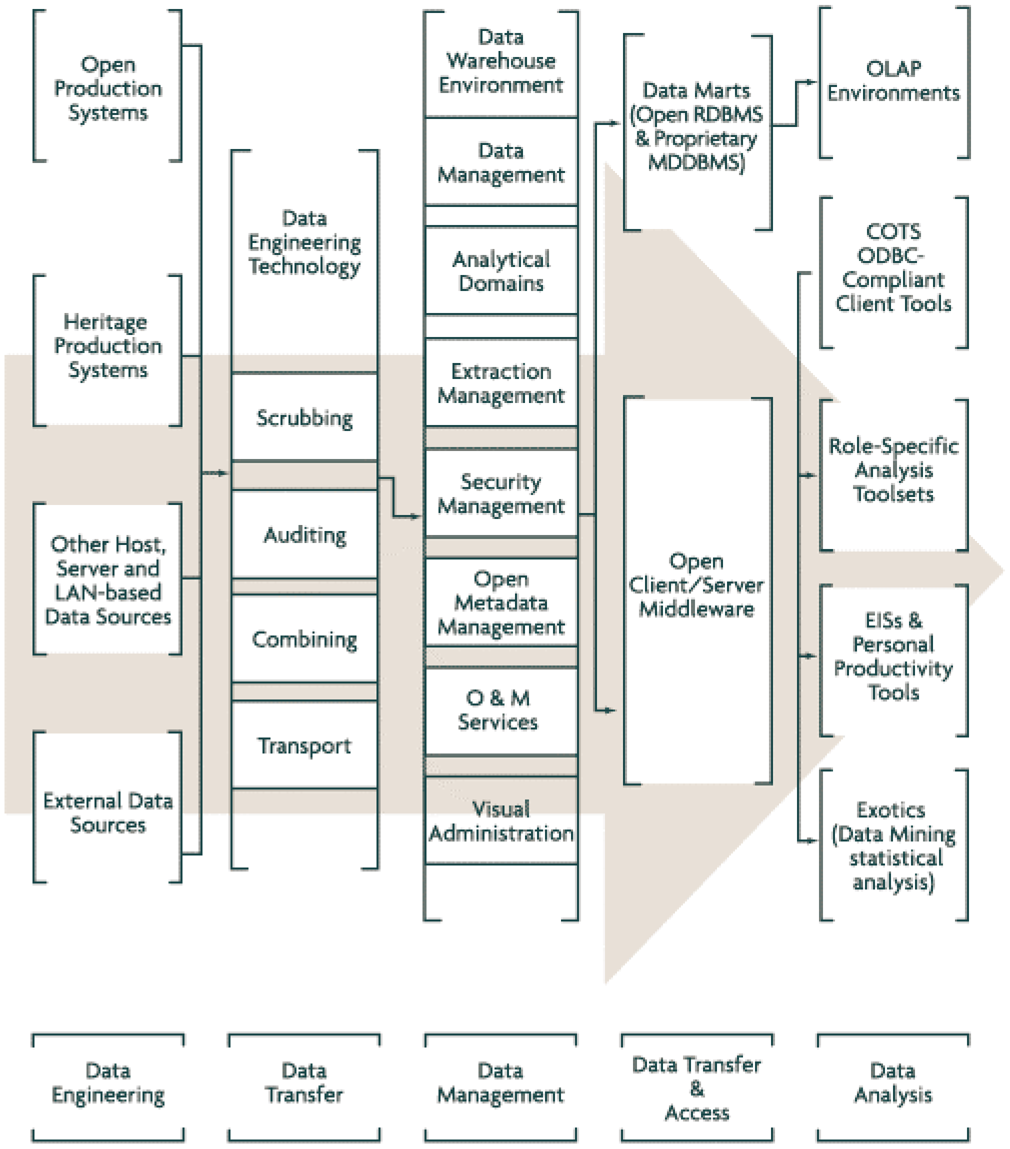
Open Production Systems
Heritage Production Systems
Other Host, Server and LAN-based Data Sources
External Data Sources
Data Engineering Technology
Scrubbing
Auditing
Combining
Transport
Data Warehouse Environment
Data Management
Analytical Domains
Extraction Management
Security Management
Open Metadata Management
O & M Services
Visual Administration
Data Marts (Open RDBMS & Proprietary MDDBMS)
Open Client/Server Middleware
OLAP Environments
COTS ODBC-Compliant Client Tools
Role-Specific Analysis Toolsets
EISs & Personal Productivity Tools
Exotics (Data Mining statistical analysis)
Data Engineering
Data Transfer
Data Management
Data Transfer & Access
Data Analysis

Figure 4 -- Warehouse/Marting Model: Software Architecture

Data Warehousing Details

The point of data warehousing is to bring together from diverse transaction-processing systems the data required by business decision makers, and store it in a single unified model of the enterprise. As such, there should be only one data warehouse for the enterprise. Because the data warehouse stores the information for the entire enterprise, it must reside at the heart of the enterprise's network environment, as close as possible to major OLTP systems, and within reach of any system that captures data of interest to end-user communities.

The data architects managing the warehouse must have the organizational and technical prowess to negotiate with each organization that contributes to the data warehouse. The data architects drive the company's fundamental business models into the data architecture, not the other way around. They also manage the data store that holds the historical data for all these business processes, and they have a unique and broad-based view -- technically and procedurally -- of the enterprise.

The data warehouse contains the enterprise's most important asset: its historic data. The EDM framework employed by the data warehouse reschematizes information from each data source into a model of all the enterprise's major processes -- manufacturing, marketing, and sales. It joins each process to other processes, eliminates duplicate information, and ensures a seamless integration of the various data streams into one legible whole.

The data loaded into the EDM is stored at the lowest level of detail, such as individual customer names, product UPC and SKU codes, individual sales orders, and invoice line items. One or more of the user communities served by the data marts will need this level of detail. Of course, no user community will need all this detailed information.

The EDM is an evolutionary model. It is not possible to construct a complete EDM and populate it immediately; rather, the EDM evolves as the business evolves, and it becomes increasingly complete as businesses contribute production data to the warehouse.

The warehouse must know how to talk to each of the production systems that feed it information, in the native language of that production system, such as SNA, DECnet, or open systems protocols such as TCP/IP. The warehouse must also know how to ask for the data it needs, using SQL or other access languages for nonrelational formats such as VSAM and QSAM. This task is assigned to the warehouse (not to each data mart) to minimize data traffic and the impact on the production systems. The warehouse must know how to respond to requests for extracts from the data marts. This offers IS the opportunity to streamline its communications network by permitting only one kind of network connection (such as TCP/IP) to the warehouse.

In designing the system, IS must also choose whether to store the data extraction instructions in the "pump" used by each data mart or in the warehouse itself. Storing the instructions in each data mart enables it to command the attention of the data warehouse, where the mart acts as the master and the warehouse as the slave. This lets divisional IS customize extracts for its data mart customers more easily.

Conversely, storing the instructions in the warehouse lets central IS control the information given to the data marts. In this case, the warehouse tells the mart when it will receive data and what that data will look like. The mart is the warehouse's slave. Where the control resides is a policy issue that an enterprise must resolve based on its approach to data management and the requirements of its end-user communities.

The reasons to limit direct access to the data warehouse are related to the costs of communications and processing power (on the warehouse platform), data security and reliability, and data availability during batch updates and extractions. Instead of giving many knowledge user communities direct access, you should give access to a small community of users including the corporate IS data architects and a subset of knowledge workers who must access vast amounts of unaggregated data. These knowledge workers may be analysts who use statistical tools to analyze or predict market trends and financial performance, managers who use complex, intelligence-augmented modeling tools to predict business unit performance based on historical data, or corporate reporting personnel.

Data Marting Details

Data marts are the "corner stores" of the enterprise, and each unique knowledge worker community has its own mart maintained by the divisional or departmental IS group. Some divisions may need only a single data mart if all knowledge workers in the division have similar information requirements. In other cases, a departmental IS organization will discover several distinct knowledge worker communities within a single department of a division.

Each data mart serves only its local community, and is modeled on the information needs of that community. For example, managers of consumer products will require different information than managers of industrial products (raw material). Consumer products have a complex competitive dimension for which syndicated market information (from companies such as Information Resources Inc. and Nielsen Marketing Research) exists, while industrial products have a simpler competitive dimension. Consumer products are sold over the counter with no advance notice of purchasing, while industrial products are sold in large lots over a longer period on the basis of existing relationships and contracts. Also, consumer products are sold through channels not controlled by the manufacturer, while industrial products are supplied directly by their manufacturers. These two communities, both composed of product managers, have different information requirements.

One of the most difficult practical problems in large organizations is drawing treaty lines between the corporate regime's IT function and the IT functions of divisions and strategic business units. Organizations can end the information wars between these regimes by placing responsibility for data marting squarely into the IT function closest to the end-user constituency. In the enterprise's information economy, corporate IS has the responsibility to "manufacture" a basic information product. Driven by their unique business goals, divisional and departmental IS organizations must enhance and repackage the basic information products manufactured by corporate IS.

Rather than combining the data management and information access roles under corporate IS, data marting gives the information delivery role to divisional or avoid a least-common-denominator approach to providing information to vastly different user communities. With data marts, divisional or departmental IS organizations take responsibility for design and implementation of data marts. They can assess their customers' needs effectively, and select the appropriate technologies based on their costs and potential benefits. For example, some divisions will have homogeneous communities of terminal users; others will have mixed communities of terminal users, intelligent desktop users, and mobile workers who need both remote and local access to their data marts.

With the knowledge of their users, a divisional or departmental IS organization can build data models in their data marts that reflect the local business model in legible schema.

In contrast, the data warehouse's UDA must enforce a global model of the enterprise by embodying key objects such as sales, markets, and customers at the lowest possible level of detail.

The data mart's primary mission is to extract the base data required by a knowledge worker community from the warehouse, transform the data, and load it into the local business model in a manner that facilitates high-performance response to end-user requests. The data warehouse is built for bulk extracts and copying data to data marts. Data marts are built for fast response to specific questions from one or many simultaneous end users. This implies that the data mart is organized and indexed differently from the warehouse, and it anticipates commonly asked questions by hard-coding the answers instead of building the answers dynamically. Also, the data mart contains the systems management and monitoring tools needed to answer the management question, "What questions are end users asking?"

Finally, too many organizations make the mistake of assuming that the key to successful enterprise DSSs is to standardize on a single toolset that locks its customers out of the rapidly developing client tool and middleware DSS markets. Data marting lets each constituency choose a toolset that is appropriate to its business team, and provides clear boundaries and interfaces with which these toolsets must conform. The mart is better able than the warehouse to handle the complexities of end-user access.

Architecture Versus Implementation: Marts Or Warehouses First?

Since 1994, when we first proposed this two-tiered architectural model, there has been a great deal of discussion on whether a firm should mart first or warehouse first. In the years since 1994, the industry has swung, each year, to one side or another of the warehousing versus marting dichotomy, caught in what we believe is a false dichotomy: namely, that a given firm must either warehouse or mart.

Professional services firms and data engineering software vendors, lured by the promise of long and lucrative contracts, are not suprisingly resolutely committed to warehousing; they castigate data marting adherents for serving "seven course meals consisting largely of Hershey bars," as Bill Inmon has recently remarked, and point out quite rightly that indiscriminate departmental marting tactics produce Hoovervilles of data marts within the firm: large numbers of rickety, poorly-designed and impossible-to-support data sources that quickly age, fail to provide the appropriate amount of detailed information and, ultimately, poison the end-user constituencies feeding from them with old, bad data.

By contrast, the OLAP community - data mart vendors all - drub the data warehousing community, quite rightly, for producing multi-year, multi-million dollar projects that create vast, detailed data junkyards that, sure enough, contain all the data any firm would ever need to make any decision, but - because of that very fact - are useful to none of the decision-making constituencies in the firm, who all get lost in the long, high and wide aisles of the dark and forbidding data warehouse.

Both sides have a point - in the abstract. Concretely, it's clear to any IS person committed to building a robust scalable decision support systems infrastructure for his or her firm that there is no such thing as the warehouse or the marts: it's a question of the warehouse and the marts, but in what order? That decision is an implementation decision, rather than an architectural one; by definition, architectural frameworks describe end-states, not implementation stages.

Here is what we can say with confidence about the criteria for making decisions about how and when to stage and deploy warehouses and marts:

1. The first-order business benefits of DSS are typically associated with providing particular user constituencies with data they need to make particular, known, critical decisions. In other words, first-order benefits to end-users are typically associated with data marts.
2. By contrast, the second-order benefits of DSS - access to all the relevant data needed to make an unforeseen decision - as well as all the first- and second-order cost savings associated with DSS are associated with the concentration of large amounts of detailed information in one location under the control and management of corporate IS. In other words, second-order benefits to end-users, and first- and second-order benefits to IS organizations, are typically associated with data warehouses.
3. The warehouse will become a necessity over time strictly to protect production systems from extract drain, as more and more production systems move to 24x7

operation, as extract windows continue to shrink, and as the demand from the data marts grows exponentially.
4. Each time a data mart is added to a DSS infrastructure without a warehouse in place, operations and management costs increase significantly, whether they are explicit - borne by the IS organization - or implicit: borne by an end-user organization.

The Optimal Strategy: Mart With Warehouse-Grade Technology

Given all this, the optimal strategy is to deploy - for a particular user constituency - a data warehouse that is partially populated, with the data required by that user constituency. Perusing this strategy allows IS organizations and their firms to get the "quick win" associated with data marts, but avoid the proprietary, non-scalable technology traditionally associated with data marts. As the user constituencies increase in number and requirements, the warehousing infrastructure can be scaled "horizontally" by adding data sets to it incrementally, and "vertically" by adding OLAP-style data marts and end-user tools "beneath" the data warehouse as the data sets in the warehouse move increasingly closer to the projected unified data architecture or enterprise data model. This is, when all is said and done, the overwhelming advantage of DSS architectures based on conventional, franchise, relational database management system (RDBMS) technology and conventional star schema DSS design: these technologies and methods can serve as the basis for both warehousing and marting.

This strategy has the additional benefit of, as MCI DSS architects describe it, enabling data mart franchising: the rapid deployment (measured in weeks) of additional data sets for new user constituencies. This franchising capability is in fact a reflection of the flexibility of the two-tiered architecture, when properly implemented.

The Coping Strategy: Mart Until The First Production System Complains

If the organization cannot build or buy a focused decision support systems solution based on data warehouse grade technology, and must instead buy an inexpensive proprietary data mart, the rule of thumb for declaring a moratorium on data mart deployment and building the data warehouse is simple: the data warehouse is a practical necessity as soon as the first production system that supplies the Hooverville of data marts runs out of time to populate the marts that require its service.
In practice, this is as likely to be a function of the limited data loading capabilities of most proprietary data mart technologies as it is a function of the extract window on the production systems, but most firms discover that, after the second or third mart, one or more of the production systems can no longer accommodate multiple requests for extracts from these marts. Typically, in Hooverville architectures, this problem is "fixed" by having the marts extract less and less often from the production systems, which ultimately has the effect of aging the data available in the marts and making decisions dependent on those marts less and less reliable and more and more dangerous.

The Failure Strategy: Build A Hooverville, Build A Data Junkyard

The one strategy that has been demonstrated not to work is the one most frequently adopted by IS organizations unable to resolve the transition from architecture to implementation. This strategy has three operant variants:

1. A "field of dreams" data warehouse, containing all available production data at the lowest possible level of detail, is built as a single tier DSS environment. It is built, but over the course of years and for millions of dollars, and ultimately, no one comes.
2. Data marts are permitted, or encouraged, to proliferate randomly, paid for off departmental budgets and administered by personnel who are nominally tasked with some aspect of business operations. Eventually a production system is fatally compromised, or network bandwidth is exhausted, or a critical decision is informed with aged data from one of the marts, and the cost of the Hooverville becomes apparent to the organization as a whole.
3. IS organizations continue to revamp their architecture and plans, but never implement any significant DSS infrastructure at either tier of the two-tiered architecture.

In the first variant, the firm incurs great cost, and no benefit. In the second scenario, any short-term first-order benefits achieved by the Hooverville of marts are erased (and then some) by the catastrophic failure of IT infrastructure and/or business decision-making processes that such a scenario inevitably leads to. In the third variant, little cost is incurred, but no benefits whatsoever are delivered.

None of these scenarios can be cast as anything but a failure, except perhaps in narrow technical terms. Deploy it how we may, what is clear is that to build a scalable, robust enterprise-grade DSS environment, a firm must both warehouse and mart, and time the deployment of components into this two-tiered architecture carefully, and with attention to the specifics of its IT infrastructure and the needs of its business decision-makers.

And What About Operational Data Stores?

Operational data stores are another architectural construct developed by W.H. Inmon. Inmon developed the notion of ODSs in response to two phenomenon we have all observed in query-intensive system use cases:

- Conventional data warehouse implementations do not handle large volumes of data (i.e., detailed data) well because of particular technology decisions made in the implementation (usually the capability of the database management system used in the implementation, or the time window for data extraction from the production systems)
- Some classes of query-intensive applications - typically those associated with front-office applications like telephone sales, outbound marketing and customer support/call centers - require detailed data on "what happened ten minutes ago?" in order to support the flows of work associated with these front-office environments.

In Inmon's architectural model, ODSs are fed, in near-real-time fashion, from production sources, and store detailed, current-valued (i.e., "now") data for query-intensive applications. Historical detail is housed in the data warehouse, not in the ODS, in this model.

ODSs make sense in query-intensive environments in which current-valued data is the currency of interaction with the customer; whether they can in fact be fed by production systems (given the current trend toward 24 X 7 operation and constant, high transaction volumes) is an implementation-dependent issue that must be addressed on a case-by-case basis.

ODSs do not make sense, any longer, as a way to avoid storing detailed data (whether current-valued or historical) in the data warehouse, or as an alternative to data warehouses, particularly since conventional relational database technology from IBM, Oracle and Informix is now easily capable of multi-terabyte storage and timely loads.

As such, ODSs should be seen as special-purpose query-intensive data stores for particular classes of front-office applications, and should be treated as peers (rather than as superiors or subordinates) of data warehouses architecturally.

It also makes sense to sever any "source-sink" connection between an ODS and a data warehouse, and to make the cost of deploying ODSs fairly dear by requiring that ODSs extract directly from production systems, and drop their data sets at the close of a refresh period, rather than transferring that data to a data warehouse. In this model, the warehouse would also be extracting data directly from production systems, in a more conventional time frame.

Conclusion: Multi-Tiered Architectures Build DSS Environments, Not Hoovervilles

The first generation of DSS models, based on mainframe technology supporting both DSSs and production systems, proved unworkable. The production systems' data models were too complex for end users to understand, and the DSS and OLTP applications had to contend for the same precious computing resources.

The second generation of DSS models also proved unworkable. Corporate IS consolidated enterprise data on single, large systems, but did not consider data legibility or end-user access. Other organizations pursued isolated, undesigned departmental marting strategies, creating network and extraction chaos and damaging the data integrity of the enterprise. The warehouse is still a data vault, and in many cases employs the same proprietary technology that continues to contribute to the lack of data-based decision making.

Data marts, on the other hand, provide business teams with useful information, but the knowledge workers see only a small segment of the company's actual business. Also, the administrative burden on corporate IS and the burden of extracting data from production systems is too great for organizations to sustain as a shantytown of data marts springs up in the corporate IT infrastructure: a Hooverville of data that is impossible to operate and maintain effectively, and that ultimately poisons the decision-making communities who use these marts, as the data sets they contain becomes less and less applicable to the decision-making processes at hand.

The third generation of DSSs emphasizes the cooperative role of the data warehouse (the corporate data store) and data marts (the corner information stores of the online enterprise), and the time is right to build this hybrid enterprise DSS model. Both in-house and consulting DSS specialists still have a lot of work to do in architecture, design, and integration, but at least they now have the tools to do it.