INTERNATIONAL REVIEW OF LAW COMPUTERS
& TECHNOLOGY, VOLUME 11, NUMBER 2, PAGES 251–261, 1997

CARFAX

The Data Mart: A New Approach to Data Warehousing

PAMELA PIPE

Introduction

Vendors have recently begun to deliver low-cost and integrated data warehouse packages intended for the rapid development of departmental data warehouses, or so-called *data marts*. The availability of these packages requires organizations to consider the role of a data mart in a data warehousing system, and whether a data mart should be built before, after, or in parallel with a corporate enterprise data warehouse. In some situations a set of distributed data marts may even eliminate the need for an enterprise-level data warehouse solution. This paper discusses the role of data marts, reviews the pros and cons of the different approaches to building a data warehousing system involving data marts, and also looks at data mart product requirements. Throughout the paper, the SmartMart package from Information Builders Inc. is used to describe the characteristics of an integrated data mart package.

Types of Data Warehouse

Data warehouses come in all shapes and sizes, but essentially they can be considered to fall into one of the following categories (see Figure 1):

- An *enterprise data warehouse* (EDW) contains integrated subject-oriented data captured from one or more operational systems or external information providers, and loaded into a separate data warehouse database. An EDW contains summarized as well as detailed data recording business operations over a period of time that may range from a few months to many years. This historical and summarized data allows detailed analysis of business activity for both tactical and strategic business decision-making.
- An *operational data store* (ODS) contains current (or near-current) detailed data for regular day-to-day business querying and reporting. As with an EDW, data in an ODS is captured from operational systems and integrated into a separate subject-oriented data warehouse. Unlike an EDW, an ODS does not contain summarized or historical data. Organizations that deploy an ODS may evolve the ODS to an EDW as they gain experience with data warehousing.

This article is based upon a paper prepared for 'Information Builders', by Colin White, DataBase Associates International, California, USA.

Correspondence: Ms Pamela Pipe, Information Builders, Wembley Point, Harrow Road, Wembley, Middx HA9 6DE, UK. < http://www.ibi.comm >

1360-0869/97/020251-11

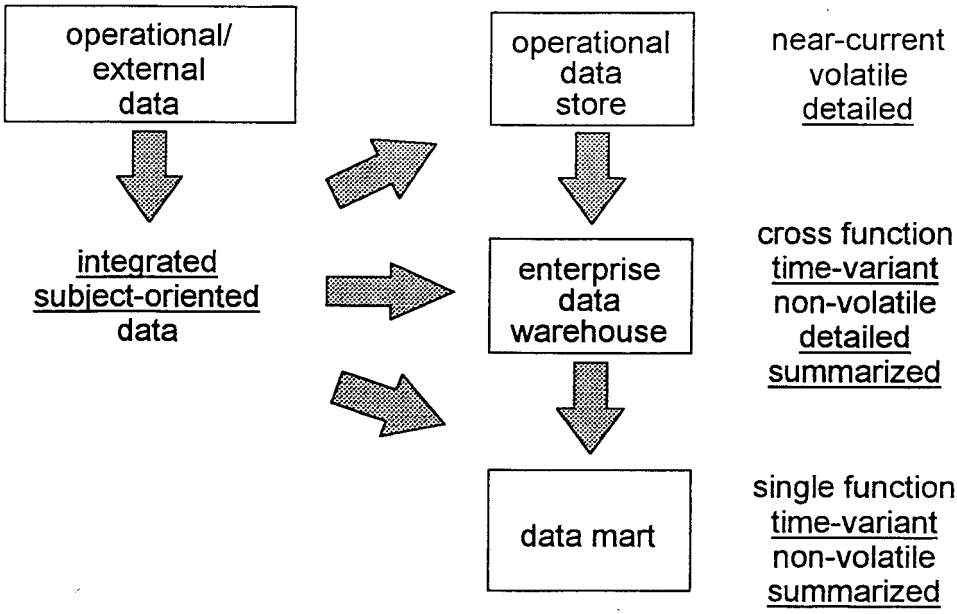

FIGURE 1. *Types of data warehouse.*

- A *data mart* contains a subset of corporate data that is of value to a specific department or set of users. This data subset may be captured from one or more operational systems, or from an EDW. Data marts usually contain summarized historical data for a specific functional area of a corporation, and have the benefit of being faster and cheaper to build than an EDW. Unlike an EDW, however, they do not provide the capability to analyze corporate data across functional areas of the business. It is important to realize that a data mart is defined by the functional scope of its users, and not by the size of the databases involved. Most data marts today involve less than 100 GB of data; some are larger, however, and it is expected that as data mart usage increases they will rapidly increase in size. Data marts can be developed prior to, in parallel with, or after the deployment of an EDW for a particular subject area. There is considerable debate about the role of data marts with respect to an EDW and how data marts should be deployed—this topic is discussed in the next section.

Approaches to Data Warehouse Development

Early proponents of data warehousing recommended building an EDW using a *top-down* approach driven primarily by the IT department. This approach involves developing a business case, assembling a project team, identifying source data of interest in operational systems, developing a data model, and then building an enterprise data warehouse. With this approach, data marts are seen as a follow-on to the construction of an EDW in a multi-tier topology (see Figure 2).

The top-down technique follows the traditional waterfall approach to IT development and has two key benefits:

- It provides a rigorous and familiar methodology for modelling and implementing end-user DSS requirements.

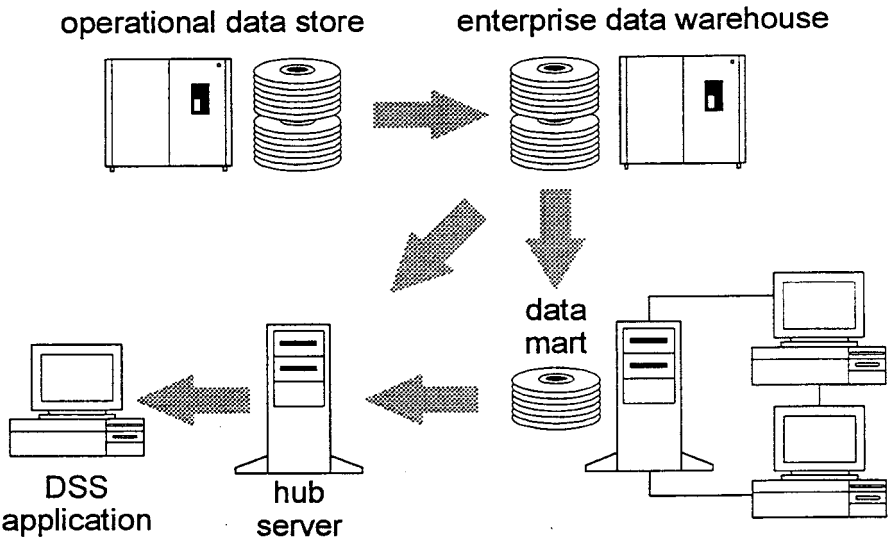

FIGURE 2. *Multi-tier data warehousing system.*

- It creates a data warehousing system that gives end users the ability to have a corporate-wide perspective of business operations, issues and opportunities for business development. A corporate-wide view of data based on a subject-oriented data model minimizes integration problems between data warehouse projects, regardless of whether they are developed serially or in parallel.

A multi-tier warehousing system involving an EDW and underlying data marts is probably the optimal one for most organizations. With this topology data marts are fed from an EDW, rather than operational systems, and data is located where it can deliver the highest availability and performance, without sacrificing integrity or control over the management of corporate data for business decision-making:

- Data marts improve availability by reducing dependency on network access to a remote EDW.
- Performance is improved by storing the data closer to the user in the data mart.
- Costs are reduced by the use of low-cost data mart software and hardware.
- Capacity planning is easier since data marts reduce EDW workload and also in some cases database size.
- Data control and integrity is maintained, since the EDW is the single source of data for feeding the data marts.

This vision of a rigorous top-down approach to data warehouse construction is not shared by everyone. Many argue that a top-down approach, where data marts are developed after an EDW is built, has several drawbacks:

- IT-driven top-down projects in the past have led to long delivery schedules, high capitalization, cost overruns, and poor end-user functionality, even though adequate cost justification was done prior to the project.
- Cost justification for a data warehouse may be somewhat elusive, since it is difficult to quantify return on investment (ROI) on a project whose major benefit is the potential for better business decision making. Executive management may find it difficult to approve such projects given the track record of IT departments.

- The current economic climate and the global nature of competition demand solutions that enable organizations not only to respond rapidly to changing business conditions, but also to be able to predict and quickly exploit new business opportunities. Data warehouse approaches that have long delivery cycles cannot deliver solutions quickly enough to address these challenges. Top-down advocates argue that an iterative approach to EDW development, where the EDW is built subject area by subject area, reduces development time. Even the most optimistic supporters of this approach, however, acknowledge that EDW development time may be closer to a year (or possibly longer).

An alternative to the top-down development of an EDW and underlying data marts is first to build the data marts, and then an EDW. Some advocates of this *bottom-up* approach even argue that an EDW is not necessary, since a corporate view of data can be achieved by combining data marts together in a *distributed* data warehouse using database *hub-server* middleware. The middleware provides a seamless interface between end user DSS tools and the data marts by insulating the user from the physical locations of the data marts. A hub server can also provide other services, for example, enhanced security, business views of data, and facilities for monitoring and controlling end-user access to data.

People who favour top-down warehouse development can rightly point out several disadvantages of the bottom-up approach:

- An uncontrolled proliferation of data marts can result in integration problems between data marts and a future EDW. Most of these potential issues would be due to differences in business terminology, data formats and representations, and required attributes not being present in the data mart data models. These problems are the primary, if not the only, objection that top-down advocates have to the implementation of data marts prior to the construction of the EDW. These problems are caused by the lack of a corporate data model for the warehouse.
- As data marts proliferate, users will want to access data marts belonging to other departments for cross-business function analysis. Seamless access to data marts may be difficult without appropriate hub-server middleware. Needless to say, such an environment is complex to administer and manage, and can also lead to poor performance when end-user queries access and join data from multiple data marts.
- Data mart technology at present is often unable to scale-up to support the increased data volumes and numbers of users that would exist in a distributed data mart environment. The current state of technology often limits data marts to less than 100GB of data and a maximum of 25 to 50 users. Of course these limits will vary by vendor product. Another important consideration here is the ability of a data mart product to be integrated with an EDW in a multi-tier data warehouse topology at some future date if so desired.

It is apparent that both the top-down and bottom-up approaches to data warehouse development have their strengths and weaknesses. What most organizations require is a data warehouse strategy that provides flexibility, low capitalization, and a rapid ROI, without unduly sacrificing control or introducing potential integration problems in the future. An ideal solution, therefore, would be a synergistic marriage of the two approaches that maximize the strengths, and minimize the weaknesses, of each approach. The *parallel* strategy suggested below uses a combination of the top-down and bottom-up approaches

and supports incremental and evolutionary data warehouse development, while at the same time attempting to solve the issues with other approaches that this paper raises:

1 Consider the use of an ODS for tactical day-to-day business decision-making when existing operational systems cannot supply integrated and consistent data, or when direct access by end users to operational data would adversely affect the performance of the operational environment.
2 Develop data marts as required for complex data analysis and strategic business decision-making by business functional area. This development should be based on a high-level subject data model, which documents the boundaries and data relationships that exist between functional areas. This data model will evolve and will form the basis for a future EDW. In fact, an EDW can be developed in parallel with the data marts using this approach. A data warehouse project team and appropriate checkpoint mechanisms should be put in place to coordinate the efforts of data mart and EDW development. The critical consideration here is the development of the high-level data model—the initial effort should not exceed more than a few months of elapsed time. The development of such a model will reduce, but not necessarily eliminate, future integration problems between projects.
3 Connect data marts together in a distributed warehouse environment using appropriate database hub-server middleware. A distributed data warehouse should be viewed as a tactical solution en route to a multi-tier data warehouse, unless usage, cost and performance considerations do not warrant an EDW.
4 Move toward a multi-tier data warehouse topology where an EDW acts as the single source of end-user data for supplying underlying data marts.

Business conditions and priorities will determine whether or not this evolutionary parallel approach is a viable solution for an organization. The features and capabilities of data mart products will also influence the success of this approach. The criterion for selecting data mart products is the subject of the rest of this paper.

Data Mart Product Requirements

A data warehousing system provides a complete end-to-end solution for supplying end users with business information. The main components of such a system consist of (see Figure 3):

- *Design tools* to design warehouse databases.
- *Source data acquisition tools* to capture data from source files and databases; and clean, enhance, transport and apply it to data warehouse databases.
- A *data manager* to manage and access warehouse data.
- GUI and Web-based *data access tools* to provide business end-users with the tools they need to access and analyze warehouse data.
- A *delivery manager* to distribute warehouse data and other information objects to other data warehouses, desktop applications, and Web servers on a corporate Intranet.
- *Middleware* to connect data access tools to warehouse databases, and the delivery manager to target systems.
- An *information directory* to provide administrators and business users with information about the contents and meaning of data stored in warehouse databases.
- *Warehouse management tools* to administer data warehouse operations.

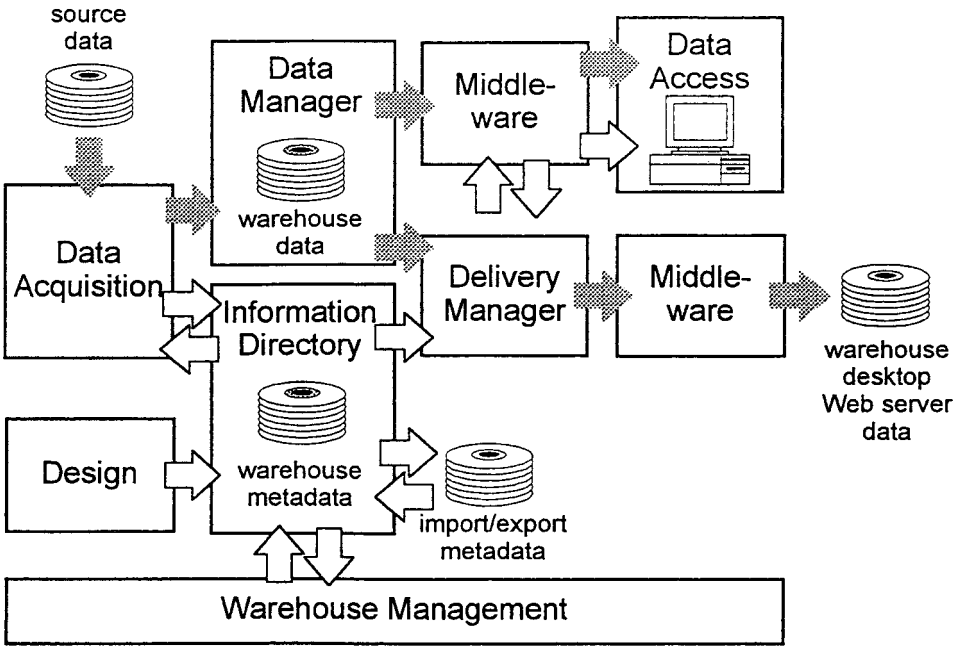

FIGURE 3. *Data warehousing system architecture.*

In addition to supplying products that support these components, vendors may also provide consulting services to train IT staff to install and deploy data warehousing products.

When building ODS and EDW systems, organizations typically select and integrate, based on application requirements, best-of-breed products for each of the components shown in Figure 3. When building a data mart, however, the cost and time to integrate best-of-breed products from multiple vendors is not acceptable, and instead organizations tend to select a data mart package (or set of integrated products) from a single vendor, which supports the components shown in the figure. In fact it is the move toward low cost and integrated data warehouse product sets that has encouraged the growth of data marts.

Potential buyers should, however, consider long-term data warehouse requirements when selecting a data mart package. Of particular importance is the ability of the data mart package to support growth as database size, numbers of users, and query volume increase with data mart usage and experience. Also of importance here with respect to future growth is the ability of the product to support distributed and multi-tier data warehousing topologies.

The next section of this paper looks at each of the components of a data warehousing system in more detail.

Design tools

Design tools are used by data warehouse designers and administrators to design and define data warehouse databases. In general, any database design tool can be used for designing a warehouse database. Compared with operational database design, there are some additional factors that need to be taken into account when designing a warehouse database, such as the handling of summary and temporal data, for example. Another factor is the use of *star schemas*, which are employed by some warehouse databases, particularly those used for multidimensional data analysis.

Source data acquisition tools

Source data acquisition tools are used to develop and run data aquisition applications that capture data from source systems for applying to warehouse databases. Data acquisition applications are developed based on *rules* defined by the warehouse developer. These rules define the data sources from which the warehouse data will be obtained, and the cleanup and enhancement to be done to this data before it is applied to warehouse databases.

Data cleanup may involve the restructuring of records or fields, removal of operational-only data, the supply of missing field values, and the checking of data integrity and consistency. Data enhancement may involve decoding and translating field values, adding a time attribute (if one is not present in the source data) to reflect the currency of data, data summarization, or the calculation of derived values.

Once the source data has been cleaned and enhanced it is mapped to the target warehouse databases, transported to the data warehousing system, and applied to the appropriate warehouse databases. The applying of data to the warehouse databases is done using data manipulation language statements (SQL, for example, in the case of a relational DBMS), or the load utility of the DBMS used to manage the warehouse.

The rules defined to data acquisition tools are often stored as *metadata* in the warehouse information directory. Some products use this metadata to generate customized 3GL/4GL data acquisition programs. Other products use this metadata dynamically during data acquisition operations to manage the flow of data from source systems to the target data warehouse.

There are four main types of product that support data acquisition:

- *Code generators* create tailored 3GL/4GL data acquisition programs based on source data definitions, and cleanup and enhancement rules defined by the warehouse developer. This approach reduces the need for an organization to write its own data acquisition programs. Most code generator products generate query language statements to capture data from the source systems. Some products also support the capturing of *changes* to source data by using, for example, the recovery log files of the source system. Code generators are used for building an EDW that acquires data from a large number of different data sources and where there is significant data cleanup to be done.
- Database *data replication tools* capture *changes* to a source database on one system and apply the changes to a copy of the source database located on a different system. These replication products often do not support the capture of changes to non-relational files and databases, and often do not provide facilities for significant data cleanup and enhancement. These tools are used to build an ODS, EDW or data mart when the number of data sources and the amount of data cleanup required are small.
- *Rule-driven data movers* capture data from the source system at user-defined intervals, clean and enhance the data, and then send and apply the results to the target warehouse database. Data to be captured from the source system is usually defined using query language statements, and cleanup and enhancement are done based on a script or function logic defined to the data pump. Depending on the product, the data mover may reside on the source system, the target system, or on a separate server. These products are used to build data marts, rather than an ODS or EDW.
- *Data re-engineering* tools are designed to perform data cleanup and enhancement. Some of these focus on structural changes to data, while others are designed to handle the cleanup of data content, such as name and address data, for example. These products are

often used in conjunction with other data acquisition tools for building an ODS or an EDW.

- *Generalized data acquisition tools and utilities* copy data from a source system to a target system. There are many tools and utilities that do not fit into any of the four categories outlined above for moving data from a source system to a target system. These products tend to focus on the fast movement of data, rather than on supporting the data integration, cleanup, and enhancement requirements of a data warehousing system.

Data manager

The data manager is used by other components in the data warehousing system to create, manage and access warehouse data (and possibly metadata). The data manager employed by a data warehousing system is usually either a relational DBMS (RDBMS) or a multidimensional DBMS (MDBMS). An RDBMS is used for building either an ODS, an EDW or a data mart, while an MDBMS is used for building a data mart for doing multidimensional analysis. The pros and cons of using an RDBMS or MDBMS for building a data mart are discussed in more detail below in the section on data access tools.

Warehouse DBMSs have requirements over and above those for operational OLTP applications. Key factors to consider here are scalability (database size, query complexity, number of users, number of dimensions, software exploitation of underlying hardware), and performance (utility operations and complex query processing). As query complexity and database size increases, data warehouse designers will need to consider the use of parallel hardware and parallel database software in order to obtain satisfactory performance.

Data access tools

Data access tools support three main user tasks: querying and reporting of known facts, analysis of known facts, and the discovery of unknown facts.

Querying and reporting involves displaying data stored in a data warehouse in a visual form. This visual form may, for example, be a printed report, or information displayed on a desktop computer. The processing may be done on-line or in batch, using *ad hoc* or pre-defined queries and reports. Tools supporting this type of task have existed for many years. Modern tools, however, often offer more sophisticated facilities like graphical user interfaces, support for Web browsers, the ability to pass retrieved data to desktop applications using techniques such as Microsoft OLE, and a semantic layer to provide a business view of the data. Query and reporting tools are most often used to provide information for tracking day-to-day business operations, i.e., for making tactical business decisions. The benefit a data warehouse offers here is data that has been cleaned and integrated from multiple operational systems.

Data analysis tools provide capabilities like mathematical and statistical functions, multidimensional modeling, and forecasting. These tools are used to analyze and forecast trends, and to measure the efficiency of business operations over a period of time. The results of this work are used for strategic business decision-making, and to look for ways of improving the efficiency of business operations and reducing costs. This type of processing is sometimes called On-Line Analytical Processing, or OLAP. Many of the tools providing this type of processing have existed for many years, but like query and reporting tools, OLAP products are now taking advantage of graphical user interfaces, Web technol-

ogy, and client/server computing. The benefit of data warehousing for OLAP is not only clean and integrated data, but also the provision of the historical data that is essential for forecasting and trend analysis.

OLAP is a hot topic, and there is much debate concerning the use of OLAP tools for doing multi-dimensional data analysis (MDA). These tools allow users to analyze and slice and dice data across multiple dimensions, for example, by time, by market, and/or by product category. Vendors offer two kinds of tool: MDA client tools that access data stored in multi-dimensional database systems (MDBMSs), and MDA client tools that access data stored in relational DBMSs. The debate centres around which type of DBMS is best suited to store and maintain multi-dimensional data. Much of what is written about the pros and cons of each approach is superficial and lacks strong technical arguments, often confusing user needs with technical issues.

When analyzing MDA products, the discussion should focus on two key, but separate, issues. The first is the requirements of the user, i.e., the functionality provided by the MDA client tool. The second issue is performance and scalability, and this is where the MDBMS vs. RDBMS issues have to be considered. There is not sufficient space in this paper to go into the arguments being put forward for each of the two MDA approaches. Suffice it to say that both approaches have some merit, and the type of MDA tool should be selected based on the kind of processing being done by the end-user and the type of data required to support this processing.

Data analysis tools typically work with summarized, rather than detailed data. These summaries can be built during analytical processing, but it is far more efficient to pre-build the summaries whenever possible. This reduces processing overheads and makes the tools easier to use. Data marts are ideally suited for the storing of summarized data that has been tailored to suit specific sets of users and applications.

Query, reporting and data analysis tools are used to process or to look for known facts, i.e. users of these tools know what kind of information they want to access and analyze. Starting to appear on the market is a new breed of business intelligence tool that is used to explore data for unknown facts, i.e. information that is not known to the business user. This style of processing allows business users to seek new business opportunities, and to look for previously unknown data patterns—to examine customer buying habits or to detect fraud, for example. Data exploration involves digging through large amounts of historical detailed data that is typically kept in an EDW. Tools that support data exploration are also sometimes called *data mining* or *data discovery* tools.

One important DSS tool direction by vendors is to add support for their use from a Web browser. The use of a Web browser to access a data mart on an internal corporate Intranet or the public Internet provides business users with a cost-effective, easy-to-use, and platform-independent data access capability.

Delivery manager

The warehouse delivery manager is used to distribute *data collections* (a data collection is a set of data of interest to a specific user or group of users) to other data warehouses and end-user applications such as spreadsheets, local databases, and so forth. The content of the data collection is usually defined by the warehouse administrator and *published* to end users in the information directory. Users *subscribe* to a collection and define a delivery schedule using the information assistant facility of the information directory. Delivery of data may be based on time-of-day or on the completion of an external event.

As experience with data warehouses increases, organizations are likely to employ data delivery approaches to supplement the facilities provided by data access tools. Of key importance here will be the ability to deliver warehouse data collections and information objects, such as reports, to business users over corporate intranets.

Warehouse middleware

Warehouse middleware provides connectivity to warehouse databases from end-user data access tools. Standard database hub server middleware can be used to perform this task, but vendors are beginning to ship specialised middleware designed for a data warehousing environment. This specialized middleware provides features such as the ability to create a single business view of data stored in multiple data warehouses (multiple, data marts, for example) and facilities to monitor, track and control access to warehouse data.

Information directory

An information directory is the roadmap to a data warehousing system—it helps users navigate their way around the information stored in a warehouse and understand the meaning of this information from a business perspective. It achieves this by providing a set of tools for integrating, maintaining and viewing warehouse metadata. In the same way as a warehouse database integrates data from multiple *data* sources, an information directory integrates metadata from multiple *metadata* sources.

The main elements of the information directory are the metadata manager, technical and business metadata, and the information assistant (sometimes called an information navigator).

- The *metadata manager* is used to maintain, export and import warehouse metadata.
- *Technical metadata* contains information about warehouse data for use by warehouse designers and administrators when carrying out warehouse development and management tasks. Technical metadata documents information about data sources, data targets, data cleanup rules, data enhancement rules, and data mapping between data sources and the warehouse databases. Most of the information is created when the warehouse designer defines the data sources and targets, and the rules to be used when copying data into the warehouse. It may also be imported from an external system, such as a 3GL copybook library, DBMS system catalogue, or a CASE tool. Technical metadata about the amount of data in the warehouse and the date it was created or updated should also be stored in the directory. Ideally this information should be added by the tools employed to capture data from operational systems and apply it to the warehouse databases. Technical metadata about how end users access and use warehouse data should also be trapped, and added to the directory to enable designers and administrators to tune and enhance the data warehouse.
- *Business metadata* contains information that gives end users an easy-to-understand business perspective of the data in the warehouse. This information includes the mapping of business subject areas to technical metadata, details about predefined queries and reports, business terms and associated technical names, and details about the custodian of the data. The business metadata is usually created by the warehouse administrator—it may also be imported from external systems, such as CASE tools or decision support tools.

- The *information assistant* provides warehouse end users with easy access to the business and technical metadata. It helps users *discover* what data exists in the warehouse and *understand* its business meaning. Some products also help users *create, document* and *run* queries, reports, or analyses, and *order* data that cannot be found in the warehouse.

In reality no vendor at present provides a complete information directory as defined above. Vendors provide instead two main types of product for maintaining warehouse metadata:

- Technical metadata repositories for managing metadata associated with the technical tasks of building a data warehousing system—database design, source data acquisition and warehouse management.
- Business information directories that focus on business metadata and support for business user access to a data warehousing system.

Warehouse Management Tools

Warehouse management tools provide a set of systems management services for maintaining the data warehousing environment. These services include managing data acquisition operations, archiving warehouse data, the backup and recovery of data, securing and authorizing access to warehouse data, and managing and tuning data access operations. At present, there are few tools designed explicitly for managing data warehousing systems, and most data warehouse administrators employ the facilities of the warehouse DBMS to perform these tasks.

Conclusion

There is more to data warehousing than just copying operational data into a separate informational database. A data warehousing system should provide a complete solution for managing the flow of information from existing corporate databases and external sources into end-user decision support systems. It should make it easy for business users to find out what information exists in the warehouse, and provide tools to access and analyze that information.

A data warehouse system may contain an operational data store or an enterprise data warehouse that manages corporate-wide information, and one or more data marts for handling departmental function-level information. The top-down and bottom-up methods of building such a system are not the most cost-effective approaches in most cases. Instead, a hybrid parallel approach to development is suggested. Regardless of which method is used, however, data marts will become a key component of data warehousing systems since they provide a cost-effective way of building data warehouses. Organizations evaluating data mart solutions should look for a single integrated package that supports the key components of a data warehousing system as outlined in this paper. Such a solution should also be capable of being integrated into a distributed data mart topology and/or a multi-tier configuration involving multiple data marts and an enterprise data warehouse.

Pamela Pipe
Wembley
UK