Data Warehouse for EIS: Some Issues and Impacts

Fred R. McFadden
College of Business Administration
University of Colorado at Colorado Springs
1420 Austin Bluffs Parkway
Colorado Springs, CO 80918 USA
e-mail "frmcfadden@.uccs.edu"

Abstract

Data warehouse is one of the most rapidly growing areas in management information systems. With this approach, data for EIS and DSS applications is separated from operational data and stored in a separate database called a data warehouse. Some of the advantages of this approach are improved performance, better data quality, and the ability to consolidate and summarize data from heterogeneous legacy systems. A data warehouse is part of a larger infrastructure that includes legacy data sources, external data sources, a repository, data acquisition software, and user interface and related analytical tools. A powerful form of data analysis, called multidimensional data analysis, is often performed by users of a data warehouse. Data warehouses can be organized into two-tier or three-tier client/server systems. Despite the complexity of the data warehouse environment, little academic research has been performed in this area. This paper identifies a number of issues that arise in the context of developing and using a data warehouse. It develops a proposed research model to determine the impact of factors such as organizational factors, warehouse infrastructure, and management support on user satisfaction and development characteristics of a data warehouse.

Keywords: *Data warehouse, executive information systems, decision support systems, multidimensional analysis, database management systems.*

Introduction

In a relatively short period of time, data warehouses have become the technology of choice for building the data management infrastructure for decision support systems (DSS) and executive information systems (EIS). In a recent survey of Fortune 1000 companies, 90% of chief information officers (CIOs) who responded to the survey claim that their organizations are developing data warehouses (Parker [23]). Data warehouse tools have evolved to the point where it is often economically feasible for even smaller firms to construct and deploy a data warehouse (Desio [4]). As a consequence, companies as diverse as Longs Drugs, Wells Fargo Bank, *The Los Angeles Times,* The Gallo Winery, and Northland Insurance have recently developed data warehouses (Kimball [19]).

The sudden popularity of data warehouses is predictable. In a typical organization, operational data are scattered throughout a variety of database management systems; in widely different formats; and on a variety of hardware platforms. Accessing this data and making it available for DSS and EIS applications is often difficult and time consuming (Rockart and DeLong [26]). The painful realization that there is not enough time or money to replace legacy systems, together with the ever-increasing demand for reliable data, has pushed data warehouse to the top of the priority list (Griffin [10]).

In simple terms, a data warehouse is a database that is optimized for decision support (Saylor and Bansal [27]). The data warehouse is normally populated through the extraction and integration of data from both operational and external data sources.

Successful data management is widely recognized as a critical success factor in developing executive information systems. In fact, poor data management has been cited as the most common cause of EIS failure (Friend [7]). A recent study identified fifteen data management issues (both managerial and technical) that impact the success of EIS (Koh and Watson [20]). Although there is much anecdotal evidence that data warehouse technology can improve data management in this context, there is little or no scholarly research at the present time to identify costs and benefits or to evaluate the various issues associated with data warehouse. The purpose of this study is to initiate scholarly research on the topic, including the development of a proposed research model for further empirical studies.

The organization of this paper is as follows. The next section presents an introduction to the data warehouse environment. The following section then discusses some key issues that arise in planning and developing a data warehouse. The next section then introduces a proposed research model, the purpose of which is to measure the impact of several variables on the success of the data warehouse effort. A concluding section is then used to summarize the issues discussed in the paper and to suggest opportunities for further research.

The data warehouse environment

As is often the case with new technology, at the present time there is no standard definition of data warehouse or of the various components associated with the data warehouse environment. However, a widely-used working definition of data warehouse is: a collection of integrated, subject-oriented databases designed to support the DSS function, where each unit of data is relevant to some moment in time. The data warehouse contains both atomic data and summarized data (Inmon [13]). A simple definition (presented in the Introduction) is: a database (or collection of databases) that is optimized for decision support (Saylor and Bansal [27]).

Warehouse architecture. A data warehouse does not exist in isolation, but is part of a larger client/server environment (Inmon [15]). Figure 1 shows the components of a typical data warehouse architecture. There are three types of components that comprise the architecture:

1. The data warehouse, and the platform and software (including the repository) that house the data warehouse.
2. The data acquisition software (or back end), which extracts data from the heterogeneous data sources (legacy systems and external data), consolidates and summarizes the data, and loads it into the data warehouse.
3. The client (or front-end) software which allows DSS and EIS users to access and analyze data in the warehouse.

Warehouse data structure. A basic premise of data warehouse is that the best solution is to copy data from operational systems and external information providers and load it into a separate integrated warehouse (White [31]). There are several reasons for a separate data warehouse;

1. Performance--the peaks and valleys of heavy DSS processing will tend to degrade the performance of on-line transaction processing systems (Bischoff [1]).
2. Data access--data required for DSS and EIS applications typically exists in a multitude of sources and data formats. Without a separate data warehouse this data will often be difficult or impossible for users to access.
3. Data formats--the data required for DSS and EIS is physically different from that used in operations. For example, EIS often requires summary data or historical data with a time dimension.
4. Data quality--operational data is often inconsistent or of poor quality. Conversion to a data warehouse provides an opportunity to apply business rules to "clean up" and reconcile this data.

Actual design of a data warehouse is based on an analysis of user requirements. The following categories of data are typically maintained in a data warehouse: current detail, historical detail, lightly summarized data, and heavily summarized data (Inmon [15]). Data in the warehouse cannot be updated by users, but is refreshed on periodic basis by data extracted from the various data sources. A facility must generally be provided for archiving aging historical detail data, to prevent unbounded growth of the warehouse.

Data in a data warehouse is most often managed using a relational DBMS. However (unlike operational systems), denormalization is often used to reduce the data to simpler structures (Kimball [19]). Indexes are generally created on all primary keys and most foreign keys to promote fast access; other indexes are created after usage patterns become established (Bischoff [1]). The relational data may be organized into two types of tables to facilitate access: fact tables, and dimension tables. This type of organization is discussed below.

Role of the repository. An active repository is a critical component of the data warehouse architecture (see Figure 1). The repository houses metadata that indicates where data comes from, how it should be translated or transformed in moving to the data warehouse, who accesses the data and how often, what business processes it drives and which critical success factors it supports (Griffin [10]). Repositories can add value to the general IS environment, but they are essential in the complex data warehouse environment.

The repository plays two distinct (but related) roles in a data warehouse environment: technical, and business-related. The technical role of the repository is to support the building and maintenance of the data warehouse. Some of the specific uses of the repository in this role are the following: document data sources and targets, capture data transformation and data cleanup rules, provide interface to CASE tools, document warehouse data model, and analyze the impact of changes (White [31], Griffin [10]).

The business-related role of the repository is to support end-users (EIS and DSS clients) in accessing and analyzing data. In this role the repository functions as an information directory (similar to a card catalog in a library) that allows users to easily navigate the complex data structures in the data warehouse. The repository is essential in supporting the powerful user interfaces that are increasingly associated with data-warehouse supported EIS. An active repository should also support user requests for adding new data to the data warehouse. For example, with one repository tool a user can request a field not available in the repository; upon finalizing the request, the tool issues a request to the mainframe legacy system to make the new information available for review (Griffin [10]).

An extended architecture. From a client/server perspective, the data warehouse architecture shown in Figure 1 is a two-tier architecture. The data warehouse and its supporting hardware and software platforms constitute a large database server that supports an enterprise-wide community of EIS ad DSS clients. The two-tier architecture appears to be the most common in data warehouse applications today. However, some organizations are evolving a three-tier architecture, with an additional server layer inserted between the data warehouse and the user community (see Figure 2).

The purpose of the new server layer in Figure 2 is to facilitate the creation of user-community specific data stores (sometimes called "data marts") that focus on end user requirements for data (Demarest [3]). For example, an organization could create an EIS server for executives, and separate DSS servers for departments or divisions such marketing, finance, and manufacturing. The data from each of these local servers is derived by extracting and/or summarizing a subset of data from the data warehouse, and is optimized for each type of user. In essence the data warehouse acts as a wholesale source of data, and that data is "retailed" to the data marts based on local need.

Data on each local server could be stored in the form of a relational database. However, when a three-tier architecture is used it is common to structure the data in the form of a multidimensional database, as indicated in Figure 2. The purpose of this approach, and its role in EIS, are described in the next section.

Developing and using a data warehouse

Developing a data warehouse means introducing a substantial amount of change and uncertainty into an organization, particularly the MIS organization. The environment of a data warehouse is very complex, requiring top management support and as much or more integration of efforts than most transaction processing systems. Data warehouse development is generally managed as an iterative process, rather than using a traditional life cycle approach (Desio [4]). In this section we describe some of the basic factors in developing and using a data warehouse.

Data warehouse development. The major steps in developing a data warehouse are described below. There is often iteration between these steps.

1. Requirements analysis. Members of the executive team are brought together to determine the kinds of information they need in an EIS to do their jobs. Depending on the scope of the warehouse, representatives from the various functional areas are also assembled to determine their needs for data in decision support systems. Specific data entities and their attributes are identified, together with sources of data (including external data). Watson and Frolick [29] have identified a portfolio of methods used by companies to determine their EIS data requirements.
2. Data model design. Based on the requirements analysis, the next step is to design a data model for the data warehouse. The data model is usually represented in the form of an entity-relationship diagram (ERD), although other representations (such as object-oriented) are also possible. The metadata associated with the ERD are housed in the repository. The data model must be validated with the participants of the requirements analysis to ensure it can answer their business questions. These persons may identify additional information they want included at this stage of the design.
3. Data mapping. This step, which is often the most complicated in the entire development process, requires mapping from legacy systems and external data sources to the new warehouse data model. Some of the issues that are addressed during data mapping include (Griffin [10]):
    a. Integration of data across applications;
    b. Reviewing and redefining legacy data elements in order to identify the set of primary elements that have unique business meaning (sometimes called *data rationalization)*;
    c. Standardization of names, formats, definitions, and translation of data codes;
    d. Synchronizing the extraction and loading of data, recognizing variations in the update cycles of source data; and
    e. Periodic summary of critical data, such as customer and product profiles.
4. Data management. This step refers to the collection, distribution, and synchronization of data in the data warehouse. Two software components are required to support this process; a collector, and a loader. The collector, driven by rules stored in the repository, selects the appropriate data from legacy systems (and from external data sources) on a periodic basis. The loader summarizes data as appropriate and stores data in the data warehouse.

Several vendors provide software tools that automate some or most of the tasks of data management. These vendors include: Prism Solutions; Evolutionary Technology, Inc.; Carleton; and Trinzic.

5. User interface. Without a powerful and intuitive interface, a data warehouse can become a "black hole" into which great quantities of data are dumped on the back end, but are never removed on the front end (Hackathorn [11]). The user interface (in cooperation with the repository) should provide features such as the following: push-button access to predefined information requests; ad hoc query tools that enable drill down and trend analysis; on-line help; and security by user, application, view, entity, or data element level.

Multidimensional analysis. A significant change is taking place in the way executives use EIS. It has been traditionally assumed that executives are primarily concerned with strategic planning and that the information they need is highly aggregated, future-oriented, relatively old, less accurate, and used infrequently (Gorry and Scott Morton [9]). However, more recent research reveals that executives today are more actively engaged in operational and management control than previously. As a consequence, executives with access to an EIS often use its drill-down capabilities (as well as other analytical techniques) to access data that are detailed, current, and accurate (Koh and Watson [20]).

To support these new requirements, a significant change is occurring in the data structures and database technology used with EIS. The most important analysis capability that has emerged for EIS (and for the data warehouse environment in general) is multidimensional analysis. In the Introduction we cited a survey indicating that 90% of CIOs claim that their organizations are developing data warehouses. Of these, 65% said that using multidimensional analysis is a high priority in their organization (Parker [23]).

Multidimensional analysis is an analytical technique that allows users to view their data in a dimensional cube (or hypercube) format, and to easily select and analyze that data. Multidimensional analysis allows end users without extensive mathematical or statistical training to perform operations such as the following: drill-down, roll-up, cross-tabulations, ratios and trends, slice and dice, and data pivoting (that is, looking at data from different perspectives). Some of the results needed by the decision maker may not be stored in the data warehouse, but are calculated dynamically from warehouse data in response to each request.

Figure 3 provides a comparison of the multi-dimensional versus relational views of data. This figure shows a data cube (a "Rubik's cube of data") with three data dimensions: PRODUCT, REGION, and QUARTER. Each dimension can have numerous sub-dimensions: For example, TIME can be have QUARTER, MONTH, WEEK, and DAY sub-dimensions. The EIS user can select and manipulate data from the cube using an intuitive, graphical interface. By visualizing the data in this format, the user can perform powerful analyses on the data much more easily than with the two-dimensional table associated with the relational format.

Another term that is sometimes used for multidimensional analysis is on-line analytical processing (or OLAP). This term, which was coined by E. F. Codd, is intended to be contrasted with on-line transaction processing (or OLTP), which is commonly used in operational systems. Codd has developed a list of 12 rules for OLAP (Frye [8]).

Multidimensional database. To provide EIS and DSS users with multidimensional analysis capability, developers must decide on how to structure the data and what type of database management system (DBMS) to use in implementing the data warehouse (for a two-tier architecture, Figure 1) or data marts (for a three-tier architecture, Figure 2). There are three basic choices, each of which we discuss briefly:

1. Use an RDBMS server. Organize the data in the form of relations (either normalized or denormalized). Provide multidimensional viewing and analysis through client interfaces and applications that interface with the RDBMS.
2. Use an RDBMS server. Organize the data in the form of a "star" structure. Provide multidimensional viewing and analysis through client interfaces and applications that interface with the star structure.
3. Use a multidimensional database (MDB) server. Organize the data physically in the form of multidimensional arrays. Provide multidimensional viewing and analysis through client interfaces that interface with the MDB.

The first option appears to be unsuitable for multidimensional analysis in many situations for two reasons: performance, and reliability. Since data stored in tabular form is not preprocessed, query responses must be computed dynamically which may slow performance. Also, incorrect responses may be generated when performing multidimensional analysis against relational data (Finklestein [5]).

With the star data structure, data are stored in a relational format that is specially designed for multidimensional analysis. Two types of tables are used: fact tables, and dimension tables. There is a separate dimension table for each data dimension: For example, for the data shown in Figure 4, there would be a TIME table, a PRODUCT table, and a REGION table. All numerical facts associated with these dimensions (for example, sales,

sales forecasts, and product returns) are stored in a single large fact table.The dimension tables contain pointers to the fact table. In most queries, the system first accesses one (or more) dimension tables, then accesses relevant data in the fact table. In essence the star structure allows the developer to represent "virtual" (rather that physical) cubes and hypercubes.

With an MDB, data are preprocessed and stored in the form of arrays (that is, in non-relational form) for fast and flexible retrieval in multidimensional analysis. Several vendors provide MDB products (also called OLAP servers) including the following: Arbor Software; Comshare, Inc.; IRI Software; and Pilot Software. These organizations have joined together to form a council (called the OLAP council) to develop industry standards for multidimensional analysis and MDB software (Frye [8]).

Some data warehouse issues

A number of issues arise in the context of developing, maintaining, and using a data warehouse and its environment. Some of these issues are technical in nature, while others are managerial. Many of the issues are the subject of industry debate today and as a consequence present rich opportunities for research.

Issue 1: Two-tier versus three-tier architecture?

The major factors that impact this issue include the size of the data warehouse, the number of actual and prospective users, and the types of analysis that are performed.

The main advantage of the two-tier architecture is its simplicity and lower cost (including fewer servers and less storage capacity). Also, development is likely to be less costly, less time-consuming, and less risky with a two-tier architecture.

The main advantages of the three-tier architecture are faster response times and the ability to custom design data for each type of user community. For example, with a three-tier architecture the data warehouse can be designed as a relational database, while an EIS server can be designed for multidimensional analysis. The question is whether the advantages of a three-tier architecture justify the additional cost and complexity. There is disagreement in the industry on this issue; one experienced industry consultant refers to the data mart concept as a "concept looking for a solution" (Kimball [19]). On the other hand, another analyst advocates a "hybrid multitiered approach" as appropriate for organizations with some or all of the following characteristics: multiple, quasi-independent lines of business or business units; a heavy investment in legacy OLTP applications, with a few legacy decision support applications; and a commitment to commonly shared business models (Demarest [3]).

Issue 2: Relational, Star, or MDB data model?

The factors that impact this issue are the size of the database, data warehouse architecture (two-tier or three-tier), the number and type of users, and the types of analysis that are performed.

The advantage of the relational data model is that it follows industry standards and is familiar to most persons in the organization. The relational model has a large number of proven query and access tools available. The relational model is well suited for query access to large quantities of data, and for applications that require updating. However, for multidimensional analysis the relational model may be difficult to use, may result in poor performance, and may be unreliable.

The advantages of the star architecture are that it is relatively simple and (since it is a variant of the relational model) it conforms to industry standards. Also, this model often provides acceptable performance for multidimensional analysis. However the star architecture may not be optimal for other types of DSS and EIS applications.

The advantage of the MDB is that it is optimized for high performance when performing multidimensional analysis. Also since data is preprocessed, the client interface is relatively simple. The major disadvantage of the MDB model is that it is a non-relational technology that is commonly implemented with proprietary DBMS products.

There is considerable industry debate at the present time concerning the appropriate choice of data models for the data warehouse environment. Although the relational model is probably most common in data warehouse applications today, all of the models described are commonly used, sometimes in combination.

Issue 3: General-purpose versus warehouse-specific DBMS?

Closely related to the choice of data model is the choice of DBMS products. General-purpose DBMS products (supplied by vendors such as Oracle, IBM, Informix, and Sybase) can be used for the data warehouse and/or the EIS/DSS server. Data warehouse technology has matured to the point that warehouse-specific DBMS products are also available for each of these purposes. Warehouse-specific products fall into two categories: relational DBMS, and multidimensional DBMS.

Since a general-purpose relational DBMS product must support transaction processing, it must have facilities for record locking, COMMITS, and other transaction-related activities. This creates additional overhead that may not be used in a data warehouse environment. A warehouse-specific DBMS product can eliminate unnecessary overhead and in addition make better use of indexes and support alternative physical organizations such as the star structure described in the previous section (Inmon [17]). Red Brick Systems is a prominent vendor of such a warehouse-specific product.

The advantage of a general-purpose relational product is that it is part of an open system that follows general industry standards. The advantage of a warehouse-specific DBMS product is that it may provide better performance and be easier to use in a data warehouse environment.

Multidimensional DBMS products are most often used in the middle tier of a three-tier architecture. These products are best at storing and analyzing aggregated data, while raw detail data is best stored in a relational database (Finklestein [5]). The major advantages of MDB products are high performance and ease of use. One disadvantage of these products at the present time is that are limited in the size of the database or the number of dimensions they can manage. Some fact tables in large data warehouses today exceed 500 million rows, which is well beyond the capacity of MDB.

Other disadvantages of MDB products are the cost of the software and training involved. Also, it has been reported that MDB products have relatively poor text handling capabilities. As a consequence, some organizations use both relational and multidimensional databases (Koh and Watson [20]). It appears that MDB products are best suited to complex multidimensional analysis applications, where the size of the database is relatively limited.

Issue 4: What is the best development strategy?

Factors that bear on this issue include the size of the company and of the data warehouse, resources available for development, and the established development process in the company.

There are two extreme approaches to developing a data warehouse. The first, a "quick and dirty" approach, involves simple replication of legacy files and tables into a warehouse. This step, which can be accomplished in a short time frame, achieves the objective of separating decision support data from operational systems. However, the data is unlikely to be useful for EIS and DSS since it retains all of the shortcomings (such as inconsistency and poor quality) inherent in the legacy systems.

At the other extreme, an organization can use a classical life cycle approach to developing a data warehouse. This approach would involve building (if they do not already exist) an enterprise business model and data model, then designing a data warehouse architecture and constructing the data warehouse. The costs and risks of this approach are well known, although it may be appropriate in some circumstances.

An intermediate (or tactical) approach is probably best for data warehouse development in most circumstances. This approach focuses on an important business issue in the organization such as financial planning or product development. A limited business model and data model are then developed related to this business issue. A prototype data warehouse is then developed and implemented. Only when the tactical warehouse is stable and users are satisfied is the warehouse expanded to include additional business areas (Brooks [2]).

Issue 5: How should the impact on MIS be managed?

Developing a data warehouse introduces a significant amount of change and uncertainty into an MIS organization. Some of the more important impacts are the following (Desio [4]):

1. The data warehouse environment (described earlier) is extremely complex, requiring more integration than with most transaction processing environments;
2. A different development methodology may be required, as described above;
3. Consistency and currency are important keys to high quality in the data warehouse. For MIS this requires increased vigilance and attention to quality;
4. Data administration and database administration play much more prominent roles in a data warehouse environment. Where these functions play a supporting role in transactional systems, they must play a central role in a data warehouse environment.

One possible response to this change is to replace the traditional functional/matrix MIS organization with a dedicated warehouse development work team. This team would contain all of the necessary skills and would be responsible for "cradle to grave" development and management of the environment.

Proposed Research Study

There is an explosion of interest in data warehouse today. Many organizations are developing data warehouses, vendors are developing a multitude of tools for the data warehouse environment, and there are numerous industry conferences and seminars addressing this topic. Nevertheless there are numerous issues, tradeoffs, and unanswered questions concerning data warehouse. Little academic research has been conducted to answer questions such as the following: What are the critical success factors for data warehouse in EIS and DSS applications? How can an organization build a business case for data warehouse? What architectural alternatives are most appropriate for a given organization?

To answer some of these questions, a research study is proposed to examine the impacts of several factors on data warehouse. The proposed research model is shown in Figure 4.

Two dependent variables are included in the model: user satisfaction with the system, and development characteristics. User satisfaction has often been used in MIS research as a surrogate measure of information system success (Yoon, Guimares, and O'Neal [32]). Components of user satisfaction in the model are: extent to which the EIS and underlying data warehouse are used for executive decision making; performance of the EIS in terms of response time; quality of the data and information provided by the system; and ease of system use. The development characteristics factor has the following components: success in meeting development schedule and cost, and extensibility of the system to accommodate new requirements.

Three independent variables are included in the study: organizational factors, warehouse infrastructure, and management support. Components of organizational factors are: strength of data and database administration, and the availability of a corporate sponsor. Components of warehouse infrastructure are: whether a data warehouse is implemented for the EIS or DSS; whether a multidimensional database is implemented; whether a true repository is implemented; and whether automated data acquisition software is used. Components of management support are: the extent to which management (including executives) participate in requirements determination; whether management provides sufficient resources for the project; and whether management helps manage organizational expectations for the data warehouse project.

Conclusions

A data warehouse is a database that is optimized for decision support. A basic premise underlying data warehouse is the separation of decision support data from operational data. This approach is often necessary in developing EIS and DSS because of the practical difficulty of accessing data from the heterogeneous data sources that exist in most organizations. Some additional benefits of data warehouse include the ability to leverage legacy systems, improved data quality, and user-friendly interfaces.

The data warehouse environment is extremely complex. It is generally implemented as a two-tier or three-tier client server environment. Data is extracted from legacy systems, consolidated and summarized, and loaded into the data warehouse. There are numerous decisions in building a data warehouse including the choice of client/server architecture, data models and DBMS that are used, and development strategies.

Little academic research has been performed on data warehouse. This paper proposes a research study and includes a data model that relates user satisfaction and development characteristic to organizational factors, the warehouse infrastructure, and management support. Many research opportunities exist in data warehouse and in EIS and DSS. Some of the issues that need to be addressed include: How should data be structured in the data warehouse?; How can multimedia data best be included and used?; What is the impact of quality of data on the usability of an EIS?; and What are the factors critical to the success of data warehouse?

References

[1] Bischoff, J., "Achieving Warehouse Success," Database Programming and Design (7:7), July 1994, p. 26-33.

[2] Brooks, P., "Tactical versus Classic Data Warehouse Development," Data Management Review (4:12), December 1994, p. 64-66.

[3] Demarest, M., "Building the Data Mart," DBMS (7:8), July 1994, p. 44-52.

[4] Desio, V., "Impact of Data Warehouse on MIS," Data Management Review (5:4), April 1995, p. 39-43.

[5] Finklestein, R., "Database Reaches the Next Dimension," Database Programming and Design (8:4), April 1995, p. 26-38.

[6] Frank, M., "A Drill-Down Analysis of Multi-dimensional Databases," DBMS (7:8), July 1994, p. 60-71.

[7] Friend, D., "Information Navigation: Blurring the Line Between DSS and EIS," DSS-89 Transactions, The Institute of Decision Sciences, 1989, p.11.

[8] Frye, C., "Big Flap over OLAP," Client/Server Computing (2:5), May 1995, p. 54-61.

[9] Gorry, G. A., and Scott Morton, M. S., "A Framework for Management Information Systems," Sloan Management Review (13:1), Fall 1971, p. 51-

70.

[10] Griffin, J., "The Role of the Repository in the Data Warehouse," Data Management Review (5:5), May 1995, p. 34-38.

[11] Hackathorn, R., "Data Warehousing Energizes your Enterprise," Datamation (41:2), February 1, 1995, p. 38-45.

[12] Haderle, D., "The Five Elements of the Data Warehouse," Data Management Review (4:5), May 1994, p. 13-19.

[13] Inmon, W., Building the Data Warehouse, John Wiley & Sons, Inc., New York, 1992.

[14] Inmon, W., "Charting the Course," Data Management Review (5:5), May 1995, p. 32-33.

[15] Inmon, W., "The Data Warehouse: Managing the Infrastructure," Data Management Review (4:12), December 1994, p. 9-13.

[16] Inmon, W., "Data Warehouse Success Requires Development Automation," Application Development Trends (2:3), March 1995, p. 41-46.

[17] Inmon, W., "Information Management: Charting the Course," Data Management Review (5:1), January 1995, p.18-19.

[18] Inmon, W., and Kelley, C., "The 12 Rules of Data Warehouse for a Client/Server World," Data Management Review (4:5), May 1994, p. 6-10.

[19] Kimball, R., "The Doctor of DSS," DBMS (7:8), July 1994, p. 54-58.

[20] Koh, C., and Watson, H., "Data Management in Executive Information Systems," (Forthcoming).

[21] Menninger, D., "Beyond EIS: Bringing Intelligence to Knowledge Workers," Data Management Review (4:12), December 1994, p. 60-62.

[22] Orfali, R., Harley, D., and Edwards, J., Essential Client/Server Survival Guide, Van Nostrand Reinhold, New York, 1994.

[23] Parker, R. S., "Enterprise Decision Support for Client/Server," Data Management Review (4:12), December 1994, p. 4-7.

[24] Radding, R., "Support Decision Makers with a Data Warehouse," Datamation (41:5), March 15, 1995, p. 53-58.

[25] Rinaldi, D. V., "Metadata Management Separates Prism from the Data Warehouse Pack," Client/Server Computing (2:3), March 1995, p. 20-23.

[26] Rockart J. F., and DeLong, D. W., "Executive Support Systems: The Emergence of Top Management Computer Use, Dow Jones-Irwin, Homewood, IL, 1988.

[27] Saylor, M. J., and Bansal, S. K., "Open Systems Decision Support: An Overview," Data Management Review (5:1), January, 1995, p. 20-24.

[28] Thé, L., "OLAP Answers Tough Business Questions," Datamation (41:8), May 1, 1995, p. 65-72.

[29] Watson, H. J., and Frolick, M. N., "Determining Information Requirements for an EIS, "MIS Quarterly (17:3), September, 1993, p. 255-269.

[30] Weil, W., "Data Warehouse Data Integrity Impact," Data Management Review (5:3), March 1995, p. 60-64.

[31] White, C., "The Key to Data Warehouse," Database Programming and Design (8:2), February 1995, p. 23-25.

[32] Yoon, Y., Guimares, T., and O'Neal, Q., "Exploring the Factors Associated with Expert Systems Success," MIS Quarterly (19:1), March 1995, P. 83-106.

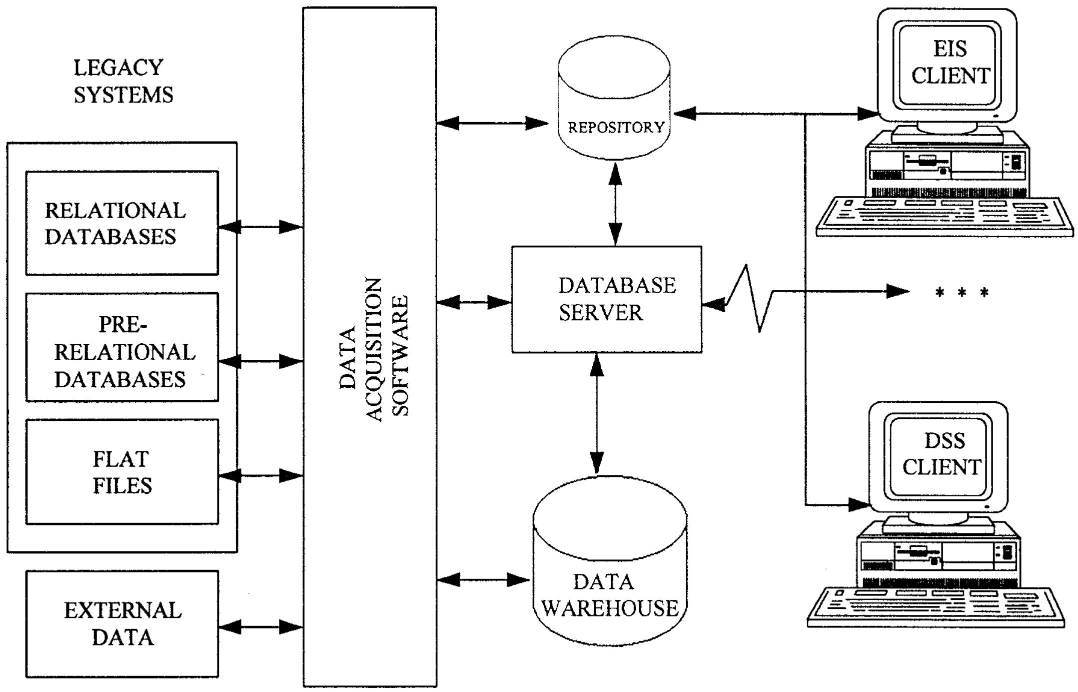


Figure 1. Data warehouse architecture (two-tier)

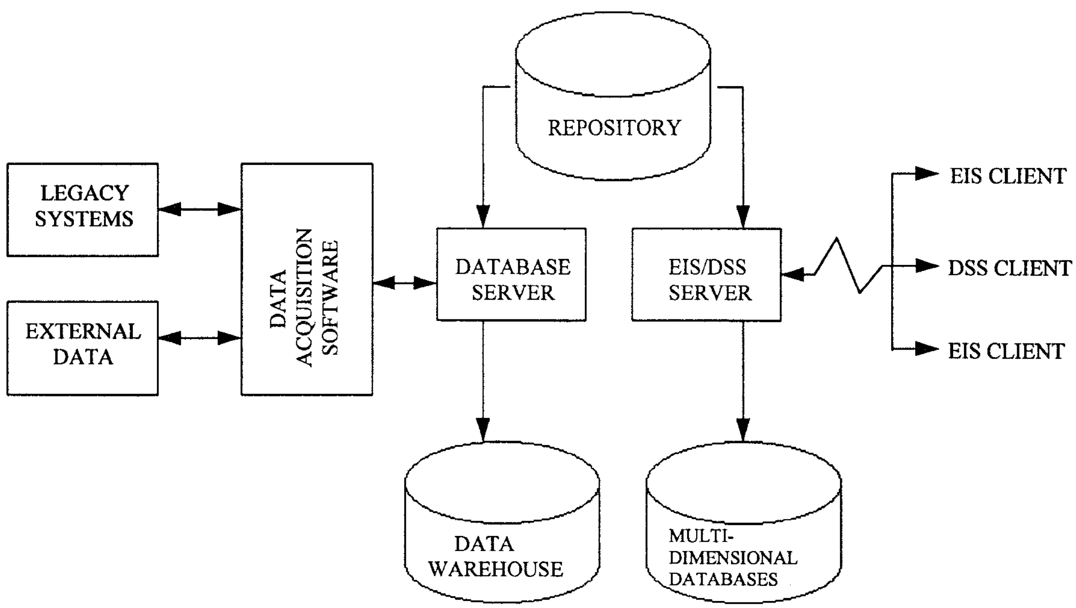
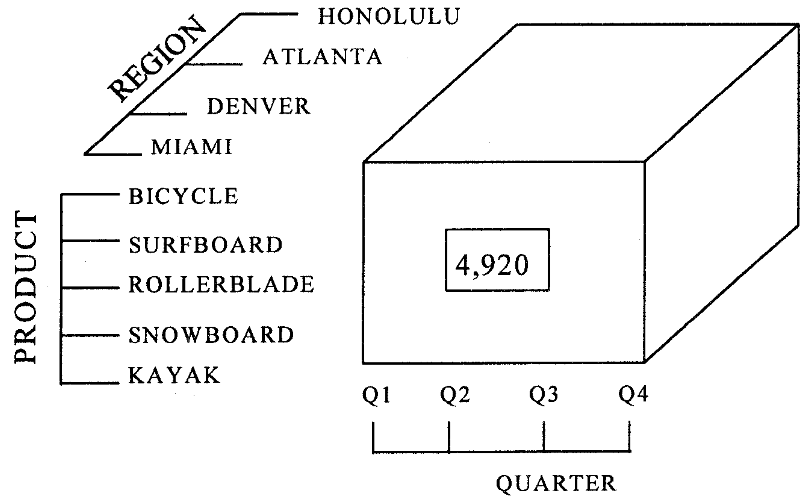


Figure 2 . Data warehouse architecture (three-tier)



PRODUCT	REGION	QUARTER	QUANTITY
BICYCLE	ATLANTA	Q1	1,562
BICYCLE	HONOLULU	Q1	3,687
SURFBRD	DENVER	Q2	129
SURFBRD	HONOLULU	Q2	8,153
SNOWBRD	HONOLULU	Q3	17
KAYAK	MIAMI	Q3	1,210

Figure 3 . Multidimensional versus relational views of data

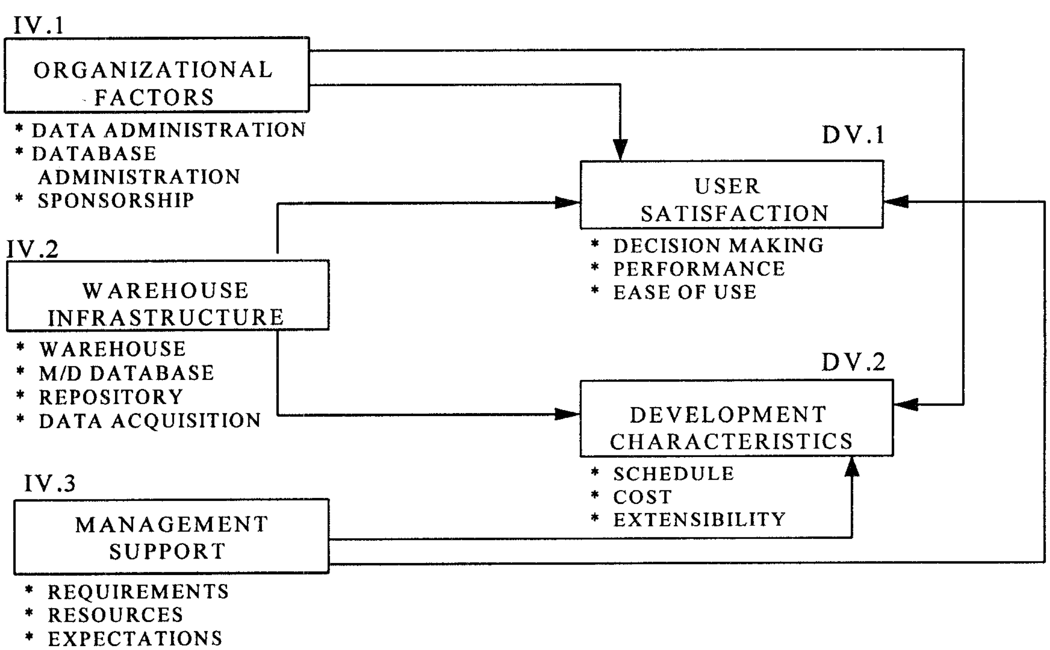


Figure 4. Proposed research model