Computer Science and Systems Analysis

Computer Science and Systems Analysis

Technical Reports

Miami University *Year* 1995

A Methodology for the Implementation and Maintenance Of a Data Warehouse

Wayne Jarrett
Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & SYSTEMS ANALYSIS

TECHNICAL REPORT: MU-SEAS-CSA-1995-006

A Methodology for the Implementation and Maintenance Of a Data Warehouse

Wayne B. Jarrett

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

A METHODOLOGY FOR THE IMPLEMENTATION AND MAINTENANCE OF A DATA WAREHOUSE

by

Wayne B. Jarrett
Systems Analysis Department
Miami University
Oxford, Ohio 45056

Working Paper #95-006 Dec. 1995

A METHODOLOGY FOR THE IMPLEMENTATION AND MAINTENANCE OF A DATA WAREHOUSE

Submitted to the Faculty of

Miami University in partial

fulfillment of the requirements

for the degree of

Master of Systems Analysis

Department of Systems Analysis

by

Wayne B. Jarrett

Miami University

Oxford, Ohio

1995

Adviser: Dr. Douglas A. Troy

ABSTRACT

A METHODOLOGY FOR THE IMPLEMENTATION AND MAINTENANCE OF A DATA WAREHOUSE

by Wayne B. Jarrett

A methodology for the implementation and maintenance of a data warehouse is described. The data warehouse forms the basis for a marketing decision support system for use by a large surgical equipment manufacturer and requires integration of multiple sources of data external to the company.

A prototype system is developed based upon the methodology. Each phase of the data warehouse implementation is discussed, including the data conversion process from raw data files to the prototype and production environments. Emphasis is placed upon the selection of suitable software tools for each process.

The research approach employed is action research, in which the researcher participates with a client organization that exhibits the problems of interest to the investigator. The client organization is a large surgical equipment manufacturer, Ethicon Endo-Surgery.

TABLE OF CONTENTS

1 Introduction

This chapter introduces data warehousing and decision support systems. The goals of the research project are presented, and the research methodology is described.

1.1 Goals

The goal of this research study is to identify a methodology for the implementation and maintenance of a data warehouse to support a marketing decision support system (DSS). A data warehouse is a read-only database of data extracted from source systems, databases, and files. Users access the data warehouse via a front-end tool or application (Poe, 1995). The data warehouse supports strategic marketing decisions using external data purchased from third party vendors supplemented by some internal data. Data from external and internal sources is transformed and integrated before loading into the data warehouse.

1.2 Research Approach

Successfully attaining the goal of this research project requires the researcher's active participation within the organization that exhibits characteristics of interest to the research. The research approach, following a model including both science and action, is action research (AR). AR involves stating an hypothesis, testing the hypothesis, and problem solving (Aguinis, 1993). Rather than just observe from the outside as is done with case studies, the researcher using the AR approach participates actively with the client organization.

Selecting from a list of project candidates at the client organization, I chose the Market Intelligence Database (MID) project. Although the MID project had been defined by the end of March 1995, when I began the research in May 1995, little development had been accomplished. Little agreement existed among the primary users of the Marketing Research department and the Information Systems (IS) staff concerning the developmental methodology. My role was clearly defined: research a methodology to implement and maintain the data warehouse, implement a prototype warehouse, and identify associated software tools.

One constraint on the research is that the client organization required adherence to systems development standards by following their own systems life cycle methodology: Ethicon Endo-Surgery System Life Cycle methodology. This methodology follows the phases of a traditional waterfall model systems development life cycle. Also, to fulfill the Food and Drug Administration (FDA) audit requirements, the methodology provides a framework for system development documentation. Another primary constraint was that the MID had to be implemented by the end of September, 1995.

Assisting my research efforts, the client organization made available to me all software and hardware tools used by the client's own systems developers.

1.3 Hypothesis

The hypothesis of the research is that a methodology can be developed to guide the development and maintenance of a data warehouse built upon external data.

1.4 Organization of the Thesis

Chapter 2 describes issues concerning data warehousing, the client organization and its requirements. Chapter 3 presents the proposed methodology. Chapter 4 discusses the prototyping system developed to implement the methodology described in chapter 3. Finally, the results of the development are described in chapter 5.

2. Background

This chapter describes data warehousing, distinguishing it from traditional databases, the major issues concerning data warehousing, and the client requirements. Issues addressed by existing data warehousing methodologies are also discussed.

2.1 Data Warehousing

Data warehousing is receiving increased attention as the trend in information technology is moving from automating existing systems to building DSS (Kelly, 1994). A data warehouse is often confused with a DSS. Silver (1991) defines a DSS as "a computer based information system that affects, or is intended to affect, how people make decisions." A data warehouse, on the other hand, is the infrastructure by which a DSS can be built (Kelly, 1994).

A data warehouse contains integrated data on a database server accessible by clients using a DSS or a query tool to process a decision support application. The data warehouse thus enables the construction of a DSS or the execution of a single query.

Data warehouses can be compared with traditional database systems. Traditional database systems concentrate upon putting the data in while data warehousing deals with getting the data out (Kimball, 1995). Traditional relational databases are designed to store enterprise business transaction data very efficiently but do not support efficient querying. Typically, data warehouses contain copies, updated periodically, of operational data and are designed to support queries. This research project concerns warehousing external data purchased from third party vendors enriched by some internal data.

2.2 Issues in Data Warehousing

Traditional relational database design centers upon the normalization of data to prevent data redundancy and update anomalies. The data model often contains many entities and understanding the relationships between them can be difficult. Data warehousing, on the other hand, contains copies of operational data made periodically to a read-only data warehouse, often in a denormalized form to facilitate querying. Denormalization does not involve the reversal of the normalization process to a lower normal form. Instead, it involves the reschematization of the data model to a simpler schema, resulting in fewer tables and a single join path to facilitate easier querying (Demarest,1994). Also, data administration becomes simpler as data relationships become easier to understand. These advantages of denormalization are relevant for primary users who want quick and simple access to data in the warehouse.

Rather than normalization, data warehousing focuses upon data integration. Data concerning a particular subject or business area is collected from different sources and

integrated into a single data warehouse to support decision making. Often, a history of data is collected to support trend analysis by the decision makers.

Maintenance of data integrity is not a primary issue because the data warehouse is read only; data integrity remains the same as in the source data. However, decision makers utilizing the data warehouse must understand that the data warehouse may not contain records of recent business transactions due to periodic updates.

Table 1 summarizes the differences between a data warehouse and a traditional database system.

Issues	*Data Warehouse*	*Traditional Database*
Normalization	Denormalized	Normalized
Update Frequency	Periodic	Continuous
Data Integration	Key issue, to support querying	Focus upon placing data in, not querying
Data Integrity	Maintained according to source data	Key issue; based upon data normalization
Data Administration	Simple, data schema easier to understand	Difficult, complex data schema
Data Sources	Multiple, includes external	Often singular, internal
Data History	Key issue, to facilitate trend analysis	Limited, older record sets archived

Table 1. Data Warehouse versus Traditional Database

Data warehouses usually contain internal operational data sometimes enriched by external data. The situation for this research project is vice-versa. It contains external data enriched by some internal data. Decision makers making tactical and strategic decisions need data from any source, internal or external, that can guide the decision making process. In this research, external data is more relevant for the decision makers due to the needs of the client organization's Marketing Research department.

Data in a data warehouse typically comes from many sources but with some common data fields. DSSs often require joins on common data fields and to permit these joins the data types must be consistent. Unfortunately, many of the common fields from the different data sources require data transformation to produce consistent data types to allow query joins.

2.3 Client Organization

Ethicon Endo-Surgery (EES), headquartered in Blue Ash, Ohio, is a wholly-owned Johnson and Johnson company. It manufactures about 480 mechanical and endoscopic surgical equipment products at plants in Blue Ash, Ohio, Albuquerque, New Mexico, and Juarez, Mexico.

The Marketing Research department, handling many new products and markets in a highly competitive environment, needs relevant industry information to determine marketing strategies. Analyzing patient, physician, and treatment data sold by third party and in-house sources, the department manipulated and presented the data manually using PC tools such as MS-Excel. It was a tedious process often combining

and manipulating data from several sources of different file formats. The time required often resulted in an incomplete analysis, limiting the support for determining marketing strategies. Thus, management defined a need to improve the process by producing a data warehouse including all the external data that the department needed to support strategic decision making. The external data would be integrated as much as possible, enriched by some internal data, and accessed by a desktop DSS.

2.4 Client Requirements

A Joint Application Design (JAD) workshop was scheduled in late May, the purpose of which was to identify all relevant issues and plan development of the MID. JAD, an IBM-developed methodology involving group development workshops, includes an executive sponsor, primary users, a facilitator, systems analysts, IS representatives, and a scribe (Porter, 1993). Facilitated by an independent consultant, the JAD workshop clearly established the user requirements, identified alternative solutions, selected a solution and scheduled the development process.

The primary requirement was that the Marketing Research department wanted the MID to permit ready access to market intelligence by primary users within their own department and secondary users from other departments. The MID was to be an Oracle 7 database located on one of the several Alpha servers accessed corporate wide via the LAN. Management saw it as the seed of warehousing external data.

Two solutions concerning access to the MID were identified during the JAD session. One solution required the construction of a DSS providing primary and secondary users access to the MID via a desktop application. The other solution did

not require a DSS application, but instead was to provide access to the MID via a query tool. This solution would require the users to develop skills using a query building tool.

A major issue addressed at the JAD session was how to control access to information in the MID. According to the primary users, establishing different data views of the MID for different users would not solve the problem. They claimed that data in the MID was easily misinterpreted by all but the sophisticated primary users, data that could affect critical strategic decisions.

Relying upon a query tool and not a DSS application was determined to be the most appropriate solution to accessing the MID. This solution would permit the flexibility that primary users wanted to extract data from the MID. It also addressed the issue of controlling access to the MID by limiting direct access to primary users only. Primary users were to be responsible for the construction and maintenance of a repository of canned reports accessible by secondary users via the LAN. This gave the primary users the opportunity to edit reports made available to the secondary users. Considering that the MID data was to be used almost exclusively by the primary users, this was not a serious limitation to secondary users.

Another significant deliverable of the JAD session was a list of identified data sources. The Marketing Research department relies heavily upon many sources of data to determine marketing strategies. Many sources of data are offered by third party vendors and, accompanying the increased demand in the health care industry, they are continually offering additional data sources.

The different data sources can be classified into four broad categories: census data, surgical procedures and diagnoses data, physician data, and hospital data. Many of these data sources have common fields including zip code, surgical procedure or diagnosis code, and metropolitan statistical area (MSA) code.

The following few examples illustrate how the data can be used. Decision makers may use census data to extract population statistics within certain zip codes to target the Medicare patient market segment. Data sources containing MSA codes may be linked to data containing zip codes via a zip code-to-MSA code conversion table. Physician data that includes surgical procedure counts makes it possible to target certain physicians when marketing new surgical devices.

2.5 Existing Data Warehouse Methodologies

Literature concerning data warehousing approaches focuses upon warehousing large amounts of internal operational data. The generic methodology includes the following phases: data extraction from source systems, data integration and transformation, loading data into a read-only database, and end-user data access via a front end tool or application (Poe,1995). Poe also describes two variations beyond the generic model: the loading of business area data warehouses from a centralized enterprise wide data warehouse and the other involves loading the data directly into business area data warehouses.

Alur also address issues concerning enterprise data warehouse development. He describes two approaches to development: top down and bottom up. The top down approach first requires the construction of a subject area enterprise data warehouse

and then gradually moving subsets of the data to business area data warehouses. The disadvantage with this approach is the lengthy delivery cycle. The bottom up approach first requires the construction of business area data warehouses and later their integration. Although this approach is likely to be more popular in practice because of the quick implementation, it may be difficult to integrate the business area data warehouses. In spite of this difficulty, it would just take too long to wait for an enterprise wide data warehouse model (Radding,1995). Another advantage offered by the bottom-up approach is that the development of smaller business area data warehouses acts as prototyping for larger enterprise data warehousing. Learning from the development of smaller business area data warehouses can be applied to integrated enterprise data warehouses.

Hackathorn presents a methodology focusing upon management of five information flows in data warehousing. His approach is to examine the dynamics of the data warehouse rather than the collection of data. The first four flows - inflow, upflow, outflow and downflow - get data in from operational systems, up to a summarized form, out to end-users, and down to archival storage. The fifth information flow is information about the data warehouse itself - meta information.

Parsaye, who is known for his Sandwich paradigm, presents another approach which accepts the probability that successful data warehousing involves iterative development. The model suggests building the meat of the sandwich first by prototyping a mini data warehouse with many of the features demanded in the production version. Revision strategies are then performed iteratively to construct the

warehouse, learning from the mistakes incurred during prototype development (Krivda,1995).

The methodologies discussed acknowledge the need to transform and integrate data from different operational sources and make the normalized operational data more legible to end-users (Demarest,1994). The purpose of data warehousing is to provide legible data to decision makers. If these decision makers misinterpret the data it will result in poor decisions.

Normalized operational data is not legible to end-users due to the many tables and many join paths between any set of tables. Demarest presents two methodologies for achieving data legibility for end users: reschematization and aliasing. Reschematization involves remodeling the operational data into a denormalized form through dimensional modeling.

Figure 1 is an example of a dimensional model which consists of a large single fact table of a business process, sales, surrounded by dimensions. Dimensions are smaller tables usually containing just a few textual fields describing the key in detail. The fact table consists of a composite key, where each part of the key relates to the single part key of a dimension, and numerical facts of the business process. In an enterprise data warehouse, separate fact tables are threaded by common dimension tables.

The dimensional model is easy for end users to understand because it attempts to mirror business processes. Querying becomes simpler because dimensional modeling

removes the multiple possible join paths in a normalized model and reduces the number of tables.

Sales Fact Table

Supplier _key
Store_key
Time_key
Product_key
sales_units
sales_dollars
cost_dollars

Supplier Dimension

supplier attributes

Store Dimension

store attributes

Time Dimension

time attributes

Product Dimension

product attributes

Figure 1. A Dimensional Model

Aliasing places a model between end-users and the operational data. It translates the illegible operational data to business models that end users can understand. It can be considered nothing more than creating another external view of the operational

databases. This strategy, therefore, does not require the construction of a separate data warehouse; it emulates a data warehouse.

Both reschematization and aliasing produce legible data to end-users so it is a cost/benefit analysis that should determine the best solution. Reshematization, and the associated data warehousing, appears to be more popular. The costs involved with building the alias model not only include building the model but also the associated demands upon existing operational database systems.

The various methodologies discussed in this section do not specifically address warehousing of external data yet present issues that must be considered in the development of the methodology proposed in this paper.

2.6 Summary

The Marketing Research department in the client organization faces a competitive environment and needs relevant industry information quickly to support strategic decision making. A data warehouse, based mainly on external industry data, offers a solution for the department's needs.

3. Data Warehouse Implementation and Maintenance Methodology

The methodology proposed for the implementation and maintenance of a data warehouse for the marketing department is presented in this chapter. The various steps of the methodology are discussed: data validation, data conversion, warehouse

creation - test environment, testing the test version, warehouse creation - production version, testing the production version, reporting and maintenance.

3.1 Validation

Validation is the process of ensuring that the raw data fields are acceptable for translation to the data warehouse table format. Issues are:

- file format
- completeness
- accuracy

Data is supplied in a variety of formats, usually as text files on floppy disks. Some vendors will only supply their data in a proprietary database with its own front end query tool. Although this may appear to prevent the use of this data in a data warehouse, data can be exported, saved as a text file, and imported into the data warehouse.

The Marketing Department had discovered that data sold by the third party vendors is rarely perfect and is often incomplete. For example, vendors often provide data with records containing only some field entries. Another large data vendor provided current physician data that included data about long time deceased physicians. The competitive nature of selling data in the health care business segment may be a major contributing factor to the incompleteness. Primary users must examine the data source thoroughly to determine its reliability and value to the decision making process. Although often imperfect, much of the data available offers

invaluable information to the Marketing Research department's strategic decision making.

One data source type likely to cause validation problems is survey data. Understanding the underlying survey questionnaire is crucial to determining the accuracy of the survey data. Since data sources often contain common data fields, cross-checking these fields is an effective method for determining data accuracy.

Only after all data has been validated by the primary users can it be included in the data warehouse. Unlike data warehouses of internal data, they are the only users within the client organization who understand the data. They are solely responsible for validating the data to be used for strategic decision making.

3.2 Data Conversion

Data conversion is the process of converting data from a raw data file to a table format suitable for the data warehouse. Issues are:

- editing
- importing

The Technical Support staff have the responsibility of performing the data conversion process. Almost all data files need some editing to permit translation to the data warehouse table format. Header information must be removed and the file cleaned up before the file is imported. Following editing, each file's data is transferred to a corresponding table in the data warehouse.

3.3 Warehouse Creation: Test Version

Technical Support staff are responsible for creating a test version of the data warehouse to facilitate testing and to fulfill client system development procedures. Issues in this phase are:

- transformation
- integration

Although each data file is translated to a table format in the data warehouse environment during the conversion phase, most tables will require some form of data transformation. Common fields in tables, such as zip code, MSA code, surgical procedure and diagnosis codes, must be of the same data type.

To facilitate querying, tables must be integrated as much as possible. In fact, to integrate the data it is best not to normalize the data. It is easier for end-users to understand a non-normalized data structure and it also improves query response times.

Leaving the data non-normalized results in some data redundancy, however, data integrity is not at risk since the data warehouse is read-only and data integrity is maintained at the same level as in the data source. Although normalizing the data would reduce the size of the database a little it is not considered to be a significant benefit.

To support query performance, commonly accessed fields are indexed. This includes not only primary keys , but also fields common to other tables.

3.4 Testing: Data Warehouse Test Version

The most detailed data warehouse testing is performed at this stage before the construction of the production version, first by Technical Support staff and then by the primary users. The Technical Support staff will conduct system construction testing involving the following tests:

- data conversion test
- data integration test

Testing the data conversion process in the test environment includes insuring that all data files are imported from the data files accurately and completely to the data warehouse environment.

Integration testing involves insuring that table manipulation did not affect data integrity, and all common data fields have the same data type. To confirm full data integration, results of queries involving all common data fields among the tables are compared with the raw data files.

Once the construction tests are complete, the primary users will perform extensive system acceptance testing which includes the following items:

- system performance
- human engineering tests
- audit testing

System performance is tested by determining response times of simple and complex queries, including during times of peak system workload. Human engineering tests, which are especially relevant to the non-sophisticated end-users, determine whether

the data warehouse system is as simple to use as expected. Audit testing confirms that the data warehouse is ready to be placed into production.

3.5 Warehouse Creation: Production Version

Once the system users have completed the acceptance testing, the DBA creates the production version of the data warehouse. Table definitions from the test environment can be used but the DBA will also require sizing information and what fields are typically joined in queries. Sizing information includes not only the current size of the data warehouse but also allowances for expected growth as new data sources are added and a history is accumulated.

The DBA uses results of the systems acceptance testing. Perhaps more fields will require indexing to improve query response time. Frequently joined tables may even be located on separate disks to facilitate responsive querying.

Populating the production data warehouse involves a simple procedure of executing SQL. Data from the test environment is copied to the corresponding production tables.

3.6 Testing: Data Warehouse Production Version

Testing the production version primarily involves systems testing by the DBA who confirms that the production data warehouse is populated accurately from the test version. Also, final acceptance testing by the primary users is performed to confirm production version performance.

3.7 Reporting

The objective of creating a data warehouse is to provide the infrastructure for DSSs. In the case of the Marketing Research data warehouse, the DSS application is the execution of queries to generate reports supporting strategic decision making.

The system solution determined at the JAD workshop permits read-only access to the data warehouse for primary users and indirect access for secondary users. Primary users using a desktop query tool will create reports for themselves and secondary users. They have the responsibility of maintaining a LAN-based repository of edited reports that provides easy access for secondary users. This solution allows primary users to control information accessed by the secondary users.

3.8 Maintenance

Responsibility of maintaining the data warehouse lies with the Technical Support staff and the DBA. Since the data is static, maintenance primarily involves adding tables to the data warehouse. Depending upon the data history requirements, updated data sources will be added as new tables, or appended as new fields to existing tables, when they become available. When the data source history is attained old tables are purged.

As validated data is received from Marketing Research, Technical Support staff will perform the data conversion process and provide table definitions for the DBA. The DBA will create the tables in the production environment and then populate them from the test environment. Primary users will update their canned queries to reflect the changes in the data warehouse.

3.9 Summary

Table 2 summarizes the methodology described in this chapter.

Steps	*Client Entities*
Validation	Marketing Services Department
Data Conversion	Technical Support staff
Warehouse Creation: Test Version	Technical Support staff
Testing: Test Version	Technical Support staff Marketing Services Department
Warehouse Creation: Production Version	DBA
Testing: Production Version	DBA and Marketing Services Department
Reporting	Marketing Services Department
Maintenance	Technical Support staff and DBA

Table 2. Methodology and the Client Organization

Implementation and maintenance of the data warehouse according to the methodology described involves three entities in the client organization: the Marketing Services department, the Technical Support staff, and the DBA. The Marketing Services department has the responsibility of providing validated data sources to the Technical Support staff who perform the data conversion process. The

DBA receives data definitions from the test environment, creates the data warehouse and then populates the tables.

4. Methodology Validation

To test the methodology and identify appropriate software tools, a prototype data warehouse was implemented. This chapter describes the prototyping technique used to implement the MID and the method described in chapter 3 is tested identifying its strengths and weaknesses.

4.1 Prototype Data Warehouse

Figure 2 is a process model of the prototype system developed to implement and maintain the MID. It indicates the processes, entities, and software tools involved in attempting to implement the methodology described in Chapter 3. Major components of the prototype system are described in subsequent sections.

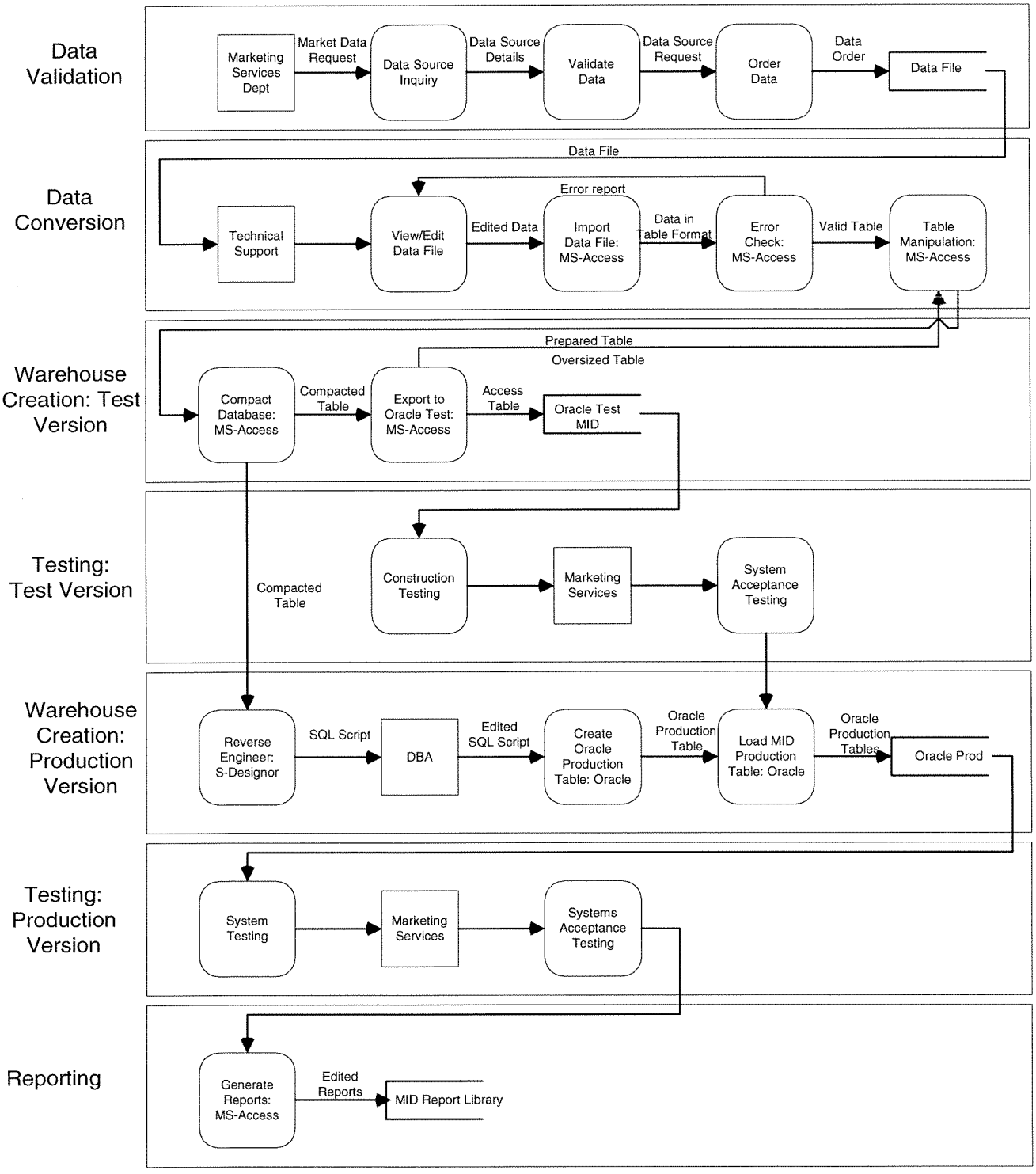

Figure 2. MID Development Data Flow Diagram

4.2 Data Validation

Data validation, a manual time-consuming process, was performed according to the methodology. Of the 28 data sources identified at the JAD session, 13 were validated and supplied in readable file formats at the time of prototyping. One data source was supplied in a proprietary database but facilitated exporting data extracts in text format.

The most effective method of determining data accuracy was cross-checking. Most data sources had some common data fields, making it possible to perform cross checking. For example, counts of certain surgical diagnoses and procedures performed at a hospital could be determined from several sources.

Many data sources contained incomplete records but this did not preclude them from the data warehouse. According to the Marketing Services department, these incomplete data sources still offered valuable information.

Validation not only involved confirming the file format and determining the accuracy and completeness but also their potential value. Some data sources were just too large. The complete census abstract, for example, is a very large set of data and most of it provides only a marginal benefit for the decision maker. Only a portion of the census data was included in the data warehouse. If the primary users need other census information they can access that information separately from the original data source.

4.3 Data Conversion

According to the client organization's requirements, the MID data warehouse was to be implemented in Oracle on a LAN-based Alpha server. To facilitate data conversion, Oracle has a utility, SQL * Loader that can read external files into an Oracle database (Oracle Corporation, 1992). However, it is not a user-friendly tool. It requires the definition of a table in the Oracle database and writing a program to import the data file contents to the defined table.

An alternative tool available to the client's systems developers, MS-Access, was selected to perform the data conversion process. A Windows based RDBMS, MS-Access offers graphical user friendliness and data manipulation utilities (Microsoft, 1994). It offers the functionality required to import files of various types easily and quickly to a database format.

Although using MS-Access to perform data conversion tasks adds complexity to the methodology described in chapter 3, it offers advantages. It saves development time and is simpler to use than the Oracle utility. MS-Access was used to perform the data importing, table manipulation, and table exporting tasks. Creating the data warehouse test version then became a more simple task than suggested by the proposed methodology. The subsequent sections describe the tools and processes used to perform the data conversion.

4.3.1 Editing

Most data files required some editing before the import process to remove header information and clean them up. Usually, they were in text format and a text editor

with the ability to handle large text files was used. Files in other formats, such as MS-Excel, were examined using the appropriate tool.

MS-Access has the ability to read the first record of a file as field titles; it was advantageous to have this information provided in files containing many fields. Some vendors offered field names in this manner upon request.

Editing often became an iterative process when MS-Access failed to transfer the file completely into a table format. In this case, MS-Access generated an error table listing the type of error and the record number in the source file. Most data import problems encountered related to parsing. For example, text fields delimited by double quotes often contained double quotes, resulting in parsing errors. To rectify this problem, the double quotes were replaced by single quotes.

4.3.2 Importing

MS-Access is a powerful graphical user-friendly tool facilitating the data import process. It can import many different file types and tables do not have to be defined in the MS-Access environment before the file is imported. During the import process, each data file is translated to a single new table or to an existing table of the correlating field format. It can read delimited text files using any delimiting character and also fixed-width text files.

Another useful MS-Access utility is the import script function. A script can be written and saved in the MS-Access database saving non-default import settings for some delimited text files and for all fixed-width text files. The script could then be recalled when importing files requiring the indentical settings.

4.3.3 Table Manipulation

All successfully imported tables required some form of manipulation, to perform data transformation, data integration and index building. MS-Access makes a best guess attempt at translating data types during the import process. Often, data types had to be changed, usually from text to number, to maintain consistency among commonly accessed fields. Fields of the same name, such as zip code, MSA code, surgical procedure code, and surgical diagnosis code, in different tables had to be the same data type to permit SQL joins.

Occasionally it was appropriate to merge tables. In one case, the data source consisted of seven small files each with the same number of records and the same large concatenated primary key. To facilitate easier querying and to reduce data redundancy, these small tables were merged into a single larger table.

Primary keys and indexes were readily built using graphical functions. MS-Access translates the graphical commands into SQL DDL, and where the translation cannot be supported, an error message was generated. For example, if a field did not contain unique values it cannot be made a primary key. However, it was indexed if it was a field likely to be accessed frequently in queries.

4.4 Data Warehouse Creation: Oracle Test

Conforming to client requirements, a test version of the data warehouse was created in the Oracle environment before creating the production version. Since the table manipulation and integration have already been performed in the MS-Access environment during the data conversion phase, creating the Oracle Test data

warehouse was fairly simple. It appeared to be just a matter of defining the tables in the Oracle Test environment and then populating them from the MS-Access tables.

MS-Access offers two choices for conveniently transferring data from MS-Access tables to a database in another environment. One method is to attach local MID tables to the Oracle tables via ODBC, and the other is to export the table objects to the Oracle MID, again via ODBC.

Before attempting the first method, the tables were defined in the Oracle Test environment. ODBC drivers facilitated the table attachment via the LAN, allowing the server based Oracle MID tables to appear local in the desktop client environment. With attachment to the unpopulated MID tables, simple SQL was used to populate the Oracle MID tables from the MS-Access environment. Although this method worked well for small tables, it was a prohibitively time consuming process, taking several hours for large tables. Often these long queries were involuntarily aborted due to a network interruption. Clearly, this method was unworkable and the alternative method was considered.

Exporting the populated tables to the Oracle MID was the better of the two alternatives. It did not require definition of the data warehouse in Oracle and took much less time than populating via attachment, taking just several minutes for even the largest table. However, due to the nature of the export process, the table objects bypassed the SQL error checking. Initially, this caused a problem when the MS-Access table and field names were expressed in lowercase because SQL Plus cannot read lowercase names. This problem was easily overcome by converting names to

uppercase in the MS-Access environment, but revealed a weakness in the Oracle error checking.

4.5 Testing: Oracle Test Version

Although, testing the data warehouse system was necessary in both the Oracle test and production environments, most of the testing activities were performed on the data warehouse test version. Results of these tests were considered when creating the production version. Testing the Oracle test version required construction testing by the Technical Support staff and systems acceptance testing by the end-users.

4.5.1 Construction Testing

Construction testing involved testing the data conversion process from raw data files to the tables in the MS-Access environment. Thorough testing proved the data conversion process successful. MS-Access had successfully detected any errors during the import process and further testing confirmed complete and accurate conversion from the MS-Access tables to the Oracle Test environment.

4.5.2 Systems Acceptance Testing

Primary users in the Marketing Services department conducted systems performance, human engineering and audit testing on the data warehouse. Query response times were faster than expected even during periods of high LAN traffic. The data warehouse passed the audit test, so now the warehouse could be placed into production.

4.6 Data Warehouse Creation: Oracle Production

Reverse engineering, a function that reads a database and produces a database script suitable for another environment, simplified the creation of the production version data warehouse. In this project it was necessary to transfer the data from the MS-Access environment to the Oracle environment. Reverse engineering the MS-Access database produced a database script for the Oracle environment.

Among the client tools available was the data modeling CASE tool, S-Designor, which supports reverse engineering of databases (SDP Technologies, 1993). The script can be for any commonly known RDBMS including Oracle 7, the target RDBMS for the MID. It offered a time saving benefit for the DBA who only had to review, and edit if necessary, the SQL script before executing it. In fact, no changes were made to the script creating the production version MID. Further, it promised a simple maintenance process for adding tables to the data warehouse.

S-Designor provides two methods for reverse engineering: via ODBC and via a database script. Unfortunately, reverse engineering via ODBC resulted in incorrect data type translation; all data types were translated as character type. S-Designor overcomes this problem by including a BASIC file that is executed within the MS-Access database to generate a script defining itself. This script was reverse engineered, producing a script to be used by the DBA to create the production MID.

Testing results of the test version were considered by the DBA. No additional indexing was needed to improve query response times, However, sizing was an important issue to be considered. Using sizing information from the Oracle Test

version and expected growth determined at the JAD workshop, the DBA allocated appropriate diskspace. To enhance query responses, tables commonly joined were placed on separate disks.

4.7 Reporting

During the prototype development, primary users learned to use the MS-Access desktop query building tool and generated reports used for strategic decision making. Although at the time of prototyping, the MID already contained about 40 tables from 13 validated data sources, most queries executed by the users were fairly simple. Many involved only 1 table and the most complex involved 3 tables and 2 joins. Following are 2 examples of typical queries made by the primary users.

First, consider the situation where only one table is accessed in the query. The user may want to know the names and medical procedure codes of all secondary procedures related to a unilateral hernia, a primary procedure. The fields of only one table, LINK_PRI_SEC, are accessed in this query and the records selected where the primary procedure is a unilateral hernia.

Second, consider a query where 3 tables are accessed requiring 2 joins. The user may want to know the total 1994 population within a given MSA code and the name and addresses of all free-standing surgical centers (FOSC) within that MSA code. This query is constructed by joining the MSA field in the CENSUS_94 and the ZIP_TO_MSA tables and joining the ZIP field in the FOSC and ZIP_TO_MSA tables. Records matching the given MSA code are retrieved showing the required field values.

4.8 Maintenance

Since it is a read-only database, maintenance of the MID is fairly simple. It involves adding new data sources and revised data from existing data sources. Thus the maintenance process involves following through all the phases of the methodology and ensuring that a data history is maintained at the required level. However, rather than adding new data as new tables, sometimes it may be appropriate to append new data fields to the existing corresponding table.

4.9 Summary

The prototype system successfully implemented the Marketing Research data warehouse. It adds complexity to the methodology proposed in chapter 3, primarily due to the user-friendly data conversion functions available in MS-Access, but nevertheless, follows the methodology. Table 3 lists the processes and tools used in developing the methodology.

Phase	*Process*	*Software Tool*
Data Validation	Data Validation	Text editor, or appropriate tool
Data Conversion	View/Edit	Text editor, or appropriate tool
	Import	MS-Access
	Table Manipulation	MS-Access
Warehouse Creation: Test Version	Export	MS-Access
Warehouse Creation: Production Version	Reverse Engineer	S-Designor
	Create Database	Oracle SQL Plus
	Populate Database	Oracle SQL Plus
Reporting	Generate Reports	MS-Access

Table 3. Development Processes and Tools Used

5. Results

This chapter discusses the results of the research to find a methodology for the implementation and maintenance of the Marketing Research Department data

warehouse. A comparison with existing methodologies, an analysis of the tools used and a discussion of user satisfaction are presented.

5.1 Comparison with Existing Methodologies

The methodology presented in this research differs from the methodologies discussed in chapter 3 because it does not involve large amounts of internal normalized operational data, but instead involves integration of large amounts of external data. The generic methodology of data warehouse construction - data extraction, data transformation and integration, loading data into a read-only database, and providing end-user access - provided a guideline for the methodology presented in this paper. However, the methodology described in this research adds to, and changes, the early phases of the generic methodology.

Data validation is an important additional phase required before external data can be loaded into a data warehouse. Also, the generic methodology's data transformation and integration becomes more expansive. Loading a data warehouse with external data is more appropriately defined as data conversion, involving the conversion of data from one of many different file types to a table format. Once the data is converted to the table format, the data is further transformed and integrated.

One of the most significant differences between the existing methodologies and that presented in this research concerns data modeling. Other methodologies require denormalization of operational data to make it more legible to end users. A data model must be prepared for the data warehouse before loading the data. This is

appropriate for the creation of data warehouses based upon internal operational data, but not for the warehousing of external market data.

For the data warehouse discussed in this research, there is no data modeling in advance of loading the data warehouse. Instead, validated data is transformed to permit integration with other tables in the warehouse.

Data from the different vendors is to be kept distinct. Combining data sources into a data table makes maintenance difficult. Vendors provide data updates periodically at different times and with differing frequency. Each data source also has its own set of characteristics which the end users have to understand to interpret the data correctly. Combining data from different data vendors would make it very difficult for the end-users to fully understand the data and therefore make good decisions.

Although the methodology presented in this research is a different from existing methodologies by adding the data validation phase and not requiring data modeling in advance, it has the following phases in common: data transformation and integration, loading data into a read-only data warehouse and providing end-user access.

5.2 Analysis of the Tools

When selecting the most suitable tools for developing the MID, the focus was not solely upon the utilities required for data conversion but also upon inter-connectivity between the MS-Access and the LAN-based Oracle databases. Microsoft ODBC drivers provide connectivity facilitating the export of database table objects from the desktop environment to the LAN-based Oracle environment. Importantly, from the

perspective of the users, ODBC also provides access to the Oracle MID via a desktop query tool.

Had it not caused translation errors, ODBC would also have been used in the reverse engineering process. MS-Access proved to be an excellent tool for developing the MID. It was the only tool available that supported user-friendly importing of external data files of various types to a database format. Permitting any form of data manipulation that was necessary to prepare the data for Oracle implementation, MS-Access offered great flexibility. Yet, it was not without disadvantages.

Unfortunately, MS-Access often locked up during complex queries such as building a single table from smaller tables. Periodically, it was necessary to compact the database to reduce the disk space it would occupy. Transparent to the user, temporary tables were built in the particular database when a query was run. These tables, although not used directly by the user, often doubled or tripled the size of the MID very quickly. A manually triggered function, the compact function, reduced the database size. However, it required an amount of free disk space equivalent to the current size of the MID.

Apart from being the most suitable tool to perform the data conversion process, MS-Access offers the facility to export the table objects directly to the MID via ODBC. It also offers an alternative method of data transfer by running SQL queries against the populated MS-Access tables and attached unpopulated Oracle tables. The

time required for this alternative was prohibitively long and often failed due to network glitches.

An interesting problem encountered during the export process occurred when first attempting to export MS-Access tables. The table objects were successfully exported into Oracle, but they were not visible to the query language. Queries could not be performed on the MID tables. This problem revealed a weakness in Oracle which, although allowed the export of MS-Access tables to the Oracle environment, could not read them. Fortunately, the problem was easily resolved. The only reason the Oracle query language could not read the tables was because the table and field names were expressed in lower case. Oracle insists that these names be in upper case. Since the export process bypassed the Oracle SQL block, this requirement escaped Oracle error detection, allowing full table definition in the data dictionary.

Another problem occurring during the export process involved exporting large tables. MS-Access had difficulty exporting tables with many fields. In this case, the large table had to be split into two smaller tables, exported and used by SQL Plus to populate the large Oracle table.

During the import process it was discovered that some data files contained too many fields to translate to a single table. MS-Access, Oracle and most RDBMSs have a maximum number of 255 fields in a table. This factor was also relevant when attempting to merge many small tables into a large table.

S-Designor, through its reverse engineering function, simplified the Oracle database creation. Generating the Oracle database script saved time; there was very little editing required of the script before it was run to create the MID.

Another important function of S-Designor is the database design documentation that it generates. This information helped fulfill the requirements of the EES System Life Cycle methodology.

5.3 User Satisfaction

During MS-Access and Oracle MID prototype testing, the primary users expressed overwhelming satisfaction with the MID. It performed according to their requirements; query execution times were well within time limit requirements established at the JAD workshop. Now, information used to develop marketing strategies is gathered within a few minutes, instead of several hours, by the execution of queries.

5.4 Conclusion

The researcher has stated and tested the hypothesis that a methodology can be developed to guide the development and maintenance of a data warehouse based on external data. The action research approach was successfully employed to develop a prototype system based on the methodology proposed.

The research employed supported the hypothesis and identified issues that are not addressed by methods in the literature. The literature did not consider the issues concerning warehousing external data: validation of the external data, conversion of

different file formats to a data warehouse table format and data modeling. The research also identified suitable tools used to implement the data warehouse.

The MID project was viewed by the client management as the seed of warehousing external data. The methodology described in the research will be used in future data warehousing, probably for data warehousing of internal operational data enriched by some external data.

A data warehousing issue not addressed by this research includes the client's approach to data warehousing. By default, the client's approach in developing the MID was a bottom-up approach rather than a top-down approach which follows a enterprise-wide data warehousing model. A major issue to consider when adopting the bottom up approach is integration of the business area data warehouses at a later date. Can the business area data warehouses, whether they are based on internal data or external data, be integrated? Future development of business area data warehouses in the client organization must consider the advantages of data warehouse integration.

The purpose of data warehousing is to support decision makers who need accurate, relevant information quickly. Strategic decision makers, who often need information from different business areas will surely benefit from integrated business area data warehouses.

References

Aguinis, H. 1993. Action Research and Scientific Method: Presumed Discrepancies and Actual Similarities, *Journal of Applied Behavioral Science,* V.29, N.4. December 1993 pp 416-431.

Alur, N. 1995. Missing Links in Data Warehousing, *Database Programming and Design,* V.8, N.9, September 1995 pp 21-23.

Demarest, M. 1994. Improving Data Legibility, *DBMS,* V.7, N.5, May 1993 pp 55-68.

Hackathorn, Dr. R. 1995. Data Warehousing Energizes Your Enterprise, *Datamation,* V.41, N.2, February 1, 1995 pp 38-43.

Kelly, S. 1994. *Data Warehousing The Route to Mass Customization,* John Wiley and Sons.

Kimball, R. 1995. The Database Market Splits, *DBMS,* V.8, N.10, September 1995 pp 12,17.

Krivda, C. 1995. Data -Mining Dynamite, *Byte,* V..20, N.10, October 1995 pp 97-102.

Microsoft, 1994. *Microsoft Access RDBMS User's Guide.*

Oracle Incorporated, 1992. *Oracle 7 Server Utility User's Guide,* chapters 4-9.

Poe, V. 1995. Data Warehouse Architecture is Not Infrastructure, *Database Programming and Design,* V.8, N.7, July 1995 pp 24-28,30,31.

Porter, J. 1993. AS/400 Information Engineering, *McGraw-Hill.*

Radding, A. 1995. Support Decision Makers with a Data Warehouse, *Datamation,* V.41, N.5, March 15, 1995 pp 53-58.

Silver, M. 1991. Systems That Support Decision-Making, *John Wiley.*

SDP Technologies, 1993. S-Designor Release 4.0 Manual, p 220.