Table of Contents

Preface 7

Acknowledgements 8

Dedication 9

Chapter 1

Data Warehousing – An Introduction 10

What is a Data Warehouse? 13

The History of Data Warehousing 14

Data Warehousing – The Driving Forces 18

Data Warehousing – The Conceptual Basis 23

Key Design Issues 47

Data Warehouse Tools 49

Data Warehouse Interfaces 52

Data Warehousing & Data Mining 53

Recommended Reading 54

Referred Standards 54

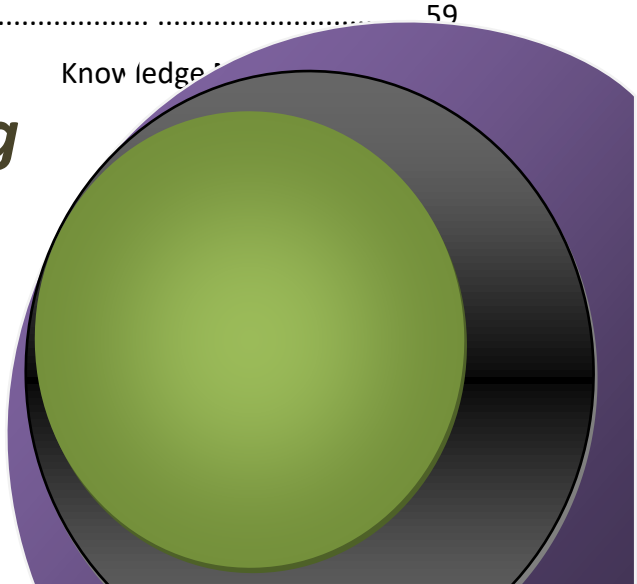
Key Terms 55

Summary 56

Check Your Learning 59

Know ledge

Data Warehousing Essentials



Chapter 2

Data Modeling & Analysis 64

Introduction 69

Data Analysis Techniques 70

Data Models & Modeling Techniques 77

Data Modeling – Key Steps 86

Data Warehouse – Structure & Composition 90

Data Warehouse Architecture 100

Data Warehouse Engineering – Life Cycle 107

Recommended Reading 119

Referred Standards 120

Key Terms 120

Summary 122

Check your Learning 124

Knowledge Map 128

List of Figures 4

Notes 5

Knowledge Maps 6

Bibliography 129

Answers to Multiple Choice Questions 131

LIST OF FIGURES

Figure 1.1 Data Warehouse - Denormalization& Transformation 29

Figure 1.2 Components of a Data Warehouse 35

Figure 1.3 Data Warehouse – Key Entities 38

Figure 1.4 Data Warehouse - Contents 40

Figure 1.5 Data Warehouse – Operational Information 42

Figure 2.1 Query & Reporting Process 72

Figure 2.2 Multi Dimensional Analysis 74

Figure 2.3 Building a Data Warehouse 89

Figure 2.4 Data Warehouse – Logical Schema 94

Figure 2.5 Data Warehouse Architecture 102

Figure 2.6 Data Warehousing Life Cycle 108

Notes

Note 1 27
Note 2 43

KNOWLEDGE MAPS

Chapter 1 63
Chapter 2 128

Preface

The deployment of Data Warehouses as a business application has grown tremendously over the past decade. Data warehouses are today considered to be one of the key component of an organizations overall IT strategy and architecture. This is especially true in the current Knowledge based global economy. Innovation and creativity is the current buzzword as business enterprises struggle to retain their stranglehold and find new markets for their products or services. Data warehouses are being developed and deployed for all businesses irrespective of its size and nature. They form the foundation over which the organizational Decision Support Systems are built. Foreseeing a huge growth potential major hardware and software vendors, across the world, have quickly developed products and services specifically targeting the data warehousing market. The objective of this book is to provide the reader with an insight to the world of Data Warehousing, in a lucid manner, devoid of mathematical complications.

Sudhir Warier

FIETE, MIMA, MISTD, [PhD], M.Phil, M.F.M, B.E

Acknowledgements

This book is a result of my nascent work in the field of Data Management, Warehousing and Mining systems, primarily in the framework design and system administration front. I have referred to multiple sources both print as well as online in preparing this manuscript. These may be credited for the success of this book while the failures are the result of my inadequacies. I have included references to these sources wherever appropriate. The presentation and the structure of this book is entirely my own.

In the ever-loving memory of my beloved father

C. V. M Warier

Chapter Objectives

The deployment of Data Warehouses as a business application has grown tremendously over the past decade. Data warehouses are today considered to be one of the key components of an organizations overall IT strategy and architecture. This is especially true in the current Knowledge based global economy. Innovation and creativity is the current buzzword as business enterprises struggle to retain their stranglehold and find new markets for their products or services. Data warehouses are being developed and deployed for all businesses irrespective of its size and nature. Foreseeing a huge growth potential major hardware and software vendors, across the world, have quickly developed products and services specifically targeting the data warehousing market. The objective of this chapter is to provide the reader with an insight to the world of Data Warehousing, in a lucid manner devoid of mathematical complications. The chapter focuses on the evolution, the organizational need and the resultant benefits in implementing a data warehousing system. The key components of a data warehousing system are also included.

KEY LEARNING'S

- Data warehousing – Evolution
- Data warehousing - Basic Concepts
- Data Warehouse Design Considerations
- Value Consistency
- Warehousing Tools – An Overview
- Data Mining – An Introduction

Chapter 1

DATA WAREHOUSING – EVOLUTION, KEY COMPONENTS & TECHNIQUES

1.0 What is a Data Warehouse?

A Data warehouse is often mistaken as a product, or a group of products that could be developed or purchased by an organization to supplement its decision-making capability. Data warehousing is not a product, or a set of products but a solution in itself. Data Warehousing is an information environment separate from the transaction-oriented operational environment and is evolving as a vital resource in the modern day knowledge enterprises. A data warehouse is a part of a universal set of processes that can help an organization in its pursuit for shoring up its decision-making capabilities. Data warehousing is the design and implementation of processes, tools, and facilities to manage and deliver absolute, opportune, accurate, and understandable information for decision-making. It includes all the activities that make it feasible for an organization to create, manage, and maintain a data warehouse.

KEY WORD - *Subject Oriented Repository*

A data warehouse manages data located outside the operational systems. A complete definition requires an understanding of many key attributes of a data warehouse system that are discussed in the subsequent chapters. A Data Warehouse is more than an archive and a means of storing and accessing corporate data. It is a subject-oriented repository designed with enterprise-wide access in mind and includes tools to satisfy the information needs of executives/managers at all organizational levels. A warehouse is not limited for complex data queries, but acts as a general facility for getting quick, accurate, and often discerning information. A Data Warehouse is designed so that its users can

recognize the information they want and access that information using simple tools. A Data Warehouse is comparable to a physical warehouse. Operational systems create data components that are loaded into the warehouse. Some of those parts are summarized into information "components" and stored in the warehouse. Data Warehouse users make requests and are delivered information "products" that are created from the components and parts stored in the warehouse. A Data Warehouse is typically a blend of technologies, including relational and multidimensional databases, client/server architecture, extraction/transformation programs, graphical user interfaces, and more. A well-defined and properly implemented Data Warehouse can be a valuable competitive tool for the knowledge enabled organizations of today and the future.

1.1 The History of Data Warehousing

The origin of data warehousing can be linked to the commercial usage of Relational database management systems (RDBMS) in starting from the early eighties. The foundation of the relational model with its simplicity, query-handling capabilities provided by the SQL language fuelled the growth of end-user computing or decision support systems (DSS).

To support end-user computing environments, data is extracted from the organizational online databases and stored in newly created systems dedicated to supporting adhoc end-user queries and multiple types of reporting functions. One of the prime concerns underlying the creation of these systems is the performance impact of this resource intensive computing on the operational data processing systems. This trepidation prompted the decision to separate end-user computing systems from transactional processing systems. In the early days the data warehouse used to contain snapshots or subsets of the operational data that were updated on a regular basis. In certain cases a limited number of versions of these snapshots were accumulated in the system while access was provided to end-users equipped with standard query and reporting tools.

The underlying data models for these DSS matched the data models of the operational systems because they were extracted snapshots. The role and purpose of data warehouses in the data processing industry have evolved considerably since those early days and are still continue to evolve rapidly. Data Warehouses are no longer identified with database systems that support end-user queries and reporting functions.

Data Warehouses should no longer be conceived as snapshots of operational data but should be considered as fresh sources of information, designed for use by the whole organization or for explicit communities of users (generally smaller in size) and data analysts within the organization. The data warehousing requirements cannot be met by employing traditional data model reengineering methodologies. It calls for an applied set of modeling techniques and a much closer interaction with the business requirements of an organization.

Data warehouses consequently acts as a source of fresh information with the objective of bringing about tangible organizational benefits. The fundamental requirements of the operational and analysis systems are different: the operational systems require high performance, whereas the analysis systems need flexibility and a broader range. It is of utmost importance to ensure that business analysis does not interfere with and degrade performance of the operational systems. The following section briefly traces the different stages in the growth of data warehousing systems.

Stage 1 - Legacy Systems

The entire system development in the early seventies was executed on IBM mainframes vide tools such as Cobol, CICS, IMS and DB2. This was followed by the deployment of mini-frame platforms such as AS/400 and VAX/VMS in the eighties, followed by the emergence of the client/server architecture. The era of distributed computing and the evolution of models to support them, starting from the early nineties, was preceded by the deployment of UNIX as a popular server platform. However inspite of all the changes in the platforms, architectures, tools,

and technologies, a large number of business applications continue to run in the mainframe environment. This is primarily due to the fact that the applications that run on legacy systems are highly difficult to migrate to a new platform.

Stage 2 - Desktop Computing

The explosive growth of personal computing or desktop computing systems in the nineties changed the IT scenario and helped introduce many new options and compelling opportunities for business analysis. The traditional gap between the programmer and the end user diminished due to the accessibility to complex analysis and graphic representation tools. Programs facilitating the extraction and processing of information from legacy systems were designed and deployed on conventional desktop environments. However a major fallback of this model of business analysis is the problem of fragmented data (single body of data split across multiple storage locations) and its consequent personalization. This is due to the fact that each individual user obtains only the information that is required by them. This process brings introduces non-standardization and renders the extracts unusable to address the requirements of multiple users. This is a major obstacle in the development and deployment of Knowledge Management Systems (KMS).

Decision-Support Systems and Executive Information Systems

The need for efficient storage of large amounts of data has been complemented by the need for performing analysis on the stored data. Decision Support Systems (DSS) and Executive Information Systems (EIS) have been designed to meet the organizational information analysis requirements. Organizational analytical requirements may vary from simple operational issues to more complex decisions involving formulating business strategies. DSS are designed to focus more on details targeted to meet the operational requirements of an organization. EIS provides a higher level of consolidation and a multi-dimensional view of the data meeting the high level organizational management requirements. One of the key features of these systems is the avoidance of cryptic technical terms and their replacement by standard descriptive business terms. The data structures of these

information systems are modeled to suit the usage requirements of non-technical users, thereby ensuring wider usage. The data is generally preprocessed with the application of standard business rules such as applicable to products, business units, and markets. Consolidated views of the data based on product, customer or markets can be made available. The systems can also be provided with drill down (focus) capability to uncover detail data. However the ability to simultaneously have access to all the detail data may not be present. These may be included via complex supplementary analytical tools. An important point to remember is that the success of data warehousing systems depends upon its alignment with the overall business structure rather than any specific requirements.

1.2 Data Warehousing – The Driving Forces

NOTE - *Key information*

In the earlier days an organization with considerable financial resources could ensure its competitive advantage due to its ability to access technological advances. The growth in the field of semiconductor technology has led to the proliferation of high power computing systems at a fraction of its earlier cost, thus bring technology within the reach of the common man. The differentiating factor in the current market driven global economies lies in the deployment of technology within an organization and the harnessing of its intrinsic knowledge (knowledge within its processes, procedures, employees, vendors, data management systems…etc.)

Driven by the need to compete more effectively, corporations are leveraging the hidden value of corporate information by making it available to the widest audience of business users through two rapidly growing technology infrastructures - the Data Warehouse and the Internet. The key to success is getting users to use the information.

TECHNOLOGY ENABLERS

There are multiple factors that have influenced the quick evolution of the data warehousing discipline. The most significant set of factors has been the explosive developments in the field of semiconductor engineering and the resultant growth of hardware and software technologies. This has contributed to sharply decreasing prices and the increasing power of computer hardware. This coupled with ease of use of the currently available software, has made possible the rapid analysis of enormous quantities of information and business knowledge. The development of processors with ever increasing capabilities and feature sets along with the growth of faster and reliable storage systems, both volatile as well as non volatile, have been the major contributors toward the growth of data warehousing systems. These developments also resulted in the birth of client/server or multi-tier computing architecture systems based on the personal computers (PC), heralding the deployment of user-friendly tools providing very simple query capabilities to integrated packages providing an incredibly powerful graphical multi-dimensional analysis tools. The resultant array of choices available for data warehouse access has contributed to its rapid evolution. The emergence of server centric Network Operating Systems (NOS) such as Windows NT and the re-emergence of Unix have brought mission-critical stability and powerful features to the distributed computing environment. The abilities and the features offered by these systems have been steadily increasing while the procurement costs have been rapidly decreasing. This has led to the introduction of intricate system features including virtual memory, multi-tasking, multi-threading and symmetric multi-processing onto the desktop computing environment

The most important development in the world of computing since the advent of the PC has been the explosion of internet/intranets and web based applications. Intranet application development has risen to become a structured and well-developed activity by itself. Intranets are private business networks that are modeled on the Internet standards. The Internet/Intranet trend has very important implications for data warehousing applications. Data warehouses can now be available worldwide on public/private network at much lower cost thus minimizing the need to replicate data across diverse geographical locations. The

development of standards also facilitates the deployment of a middle tier where all the analysis takes place before it is presented to the web-browsing client for use. The increased computing power along with the availability of affordable and point-and-click reporting and analysis tools have played an important role in evolution of data warehouses.

Another key factor that has contributed to the developments in the field of data warehousing has been the development, deployment and the increasing use of business application suites by SAP AG, Baan, Oracle, PeopleSoft among other developers. This has contributes to the introduction and subsequent growth of multi-tiered application development architecture. These applications are replacing the custom developed legacy applications of yester years. These applications would be a primary data source for an organizational data warehouse. The development of standard application programming interfaces (API) as well as migration tools has simplified the process of porting the data from diverse organizational packages. These standard applications have extensive customization features as a result of which data acquisition from these applications can be much simpler than from the earlier mainframe systems.

ECONOMIC ENABLERS

A significant influence on the evolution of the data warehousing science is the fundamental changes in the twenty-first century business organization, structure and culture. The emergence of a vibrant global knowledge based economy has had a profound impact on the information demands made by organizations worldwide. Organizations irrespective of its origins, size or sector have found markets for their products globally while competing with other business entities in vastly different cultures and economic environments. In away the economic recession witnessed across the globe in the late eighties and the early nineties contributed to the consolidation of multiple global businesses. The emergence of a global market forced organizations to reevaluate their business practices and the subsequent emergence and application of reengineering methodologies like Business Process Re-engineering (BPR). A considerable amount of effort and time

were spent by organizations to identify their core competency areas and hive of non-profitable offshoots. The traditional competitive advantages enjoyed by large organizations, on account of access to the latest technological advantages, were negated by the explosive developments in the field of semiconductor technology and allied fields. This brought high end computing environments within the reach of small size businesses and the common man, thereby creating an entirely different competitive entity. Organizations had to rapidly evolve and change as per the prevailing market dynamics. These factors immensely contributed to the rapid developments in the fields of data warehousing and data mining. The banking industry has been a pioneer in the deployment and use of data warehousing technology. Organizations in India were a little slow in recognizing the potential of warehousing and mining systems, but having done so are making rapid progress in their deployment. One of the contributory factors has been the emergence of the Business Process Outsourcing (BPO) model, resulting in migration of manufacturing and the service industry to developing countries.

The modern day data warehousing systems are extensively used for increasing organizational profitability as well as customer behavior analysis. The emergence of this global economy has led to the migration of manufacturing industries to less expensive and less restrictive countries (BPO). These spurt of opportunities presented a very volatile business climate and economies that are impossible to fathom. Business enterprises have begun to focus on building of products that can sell worldwide and in the process have also changed their strategy to sell products in the emerging global markets.

This globalization of business has increased the need for a more continuous analysis and centralized management of data. The process of collating data from far-flung business units has now started to impact a larger number of corporations. Globalization of business has made the consolidation of data in a central data warehouse more complicated. Factors such as currency fluctuations and product customization for different markets have added complexity to data warehousing, making the analysis much more complicated.

END USER PROLIFERATION

Many factors affect the heightened awareness of trends in information technology (IT) among mid and upper management levels. IT is now a universally accepted key strategic business asset and technology enabler. The explosive use of internet has greatly aided in the awareness of technology trends. The Internet is now being used to conduct business transactions; but its greatest asset to this date has been dissemination of information. Present day executives can review various industry trends and readily find case studies and vendor information online. The use of technology by mid and upper level managers has increased significantly beyond conventional email usage. This hands-on use of information and technology especially the decision making hierarchy or the top management within an organization has facilitated the sponsorship of larger projects such as data warehousing. Alongside the availability of key enabling technologies, these fundamental changes in the nature of business over the past decade have played a central role in the evolution of data warehouse.

These factors have contributed to the evolution of a technology-savvy business analyst. These technology-savvy end users play an important role in the development and deployment of data warehouses and form the core users who demonstrate the initial benefits of data warehouses. These end users are also critical to the development of the data warehouse model. Word processing and spreadsheets were the first applications to be effectively used on the PC's. The spreadsheet along with its charting functions represents one of the most extensively used business analysis and presentation functions. The new pivot tables available in popular spreadsheets have allowed for simple multi-dimensional analysis. The aggressive use of inexpensive personal productivity software has led to use of more robust reporting and analysis tools along with more powerful desktop database engines. These powerful tools are now more targeted towards the end user and often require very little training for simple applications.

1.3 Data Warehousing - The Conceptual Basis

Having detailed the evolution, need and the benefits of data warehousing in the preceding sections, we will now proceed to understand the basic terminologies and the conceptual basis for the design of data warehousing systems.

NOTE - *Key information*

Based on a detailed consideration of the various attributes and functions, a data warehouse can be broadly defined on the following lines: A data warehouse is a structured extensible environment, continuously updated and maintained for a length of time, designed for the analysis of non-volatile data [logically and physically transformed from multiple source applications] and expressed in simple business terms in alignment with the organizational business structure.

The primary concept of data warehousing is that the data stored for business analysis can most effectively be accessed by separating it from the data in the operational systems. Many of the reasons for this separation have evolved over the years. In the past, legacy systems archived data onto tapes as it became static or obsolete and many analysis reports ran from these tapes or mirror data sources to minimize the performance impact on the operational systems. These reasons to separate the operational data from analysis data have not significantly changed with the evolution of the data warehousing systems, except that now they are considered more formally during the data warehouse building process. Advances in technology and changes in the nature of business have made many of the business analysis processes much more complex and sophisticated. In addition to producing standard reports, today's data warehousing systems support very sophisticated online analysis including multi-dimensional analysis.

The most important reason for separating data for business analysis from the operational data has been the potential performance degradation on the operational system that can result from the analysis processes. High performance and quick response time is almost universally critical for operational systems. The loss of efficiency and the costs incurred with slower responses on the predefined

transactions are usually easy to calculate and measure On the other hand, business analysis processes in a data warehouse are difficult to predefine and they rarely need to have rigid response time requirements.

KEY ATTRIBUTES

- Data Characteristics
- Operational Terms
- Attribute definition
- Value Consistency
- Physical Model
- Logical Model
- Data Storage
- Data Transformation
- Data Summarization
- Views

For an operational system, it is typically possible to identify the mix of business transaction types in a given time frame including the peak loads. It is also relatively easy to specify the maximum acceptable response time given a specific load on the system. The cost of a high response time can be computed by considering factors such as the cost of operators, telecommunication costs, and the cost of any lost business. For example, an order processing system might specify the number of active order placed as well as the average placement per hour. Even the query and reporting transactions against the operational system are most likely to be predefined with predictable volume.

Even though many of the queries and reports that are run against a data warehouse are predefined, it is nearly impossible to accurately predict the activity against a data warehouse. It is common to have adhoc queries in a data warehouse that are triggered by unexpected results or by user's lack of understanding of the data model. Further, many of the analysis processes tend to be all encompassing whereas the operational processes are well segmented. A

user may decide to explore detail data while reviewing the results of a report from the summary tables. After discovering certain interesting sales activity in a particular month, the user may explore the activity for the current month, in relation to the marketing activities conducted during the month to understand the sales pattern for a particular region. Of course, there would be instances where a user attempts to run a query that will try to build a temporary table that is a cartesian product of two tables containing a million rows each. While an activity like this would unacceptably degrade an operational system's performance, it is expected and planned for in a data warehousing system. Following are some of the key attributes of a data warehouse.

1. DATA CHARACTERISTICS

A key attribute of the data within a data warehouse system is that it is loaded on to the warehouse after it has become non-volatile. This means that after the data is in the data warehouse, there would be no modifications to be made to this information. For example: After the placement of an order, its status would not change (not the delivery status), the inventory snapshot does not change, and the marketing promotion details do not change. This attribute of the data warehouse has important implications for the kind of data that is brought to the data warehouse and the timing of the data transfer. In an operational system the data entities go through many attribute changes. For example, an order may go through many stages before it is completed, or a product flowing through a conventional assembly line would have multiple processes applied to it. In general the data from an operational system is triggered to go to the data warehouse when most of the activity on these business entity data has been completed. This may mean completion of an order or final assembly of an accepted product. Once an order is completed and shipped, it is unlikely to go back to the initial status, or once a product is developed, it is unlikely to go back to its initial production stage. Another important example can be the constantly changing data that is transferred to the data warehouse one snapshot at a time. The inventory module in an operational system may change with nearly every transaction. It is therefore impossible to

carry all of these changes to the data warehouse. Depending upon the business requirements it is quite possible that a snapshot of inventory carried once every week to the data warehouse is adequate for all analysis. This would imply that the snapshot data is non-volatile.

Note - 1

An important point to note is that once the data is loaded onto a warehouse, it is not supposed to be modified. It is very difficult to maintain dynamic data in the warehouse. Any attempt to synchronize volatile data between operational and data warehousing systems will fail.

De-normalization

De-normalization is an important process in data warehousing modeling due to the fact that a relationship between many attributes does not change in this historical data.

Example - 1

For example, in an operational system, a product may be part of the product group “A” in a current month and product group “B” from the subsequent month. In a properly normalized data model, it would be inappropriate to include the product group attribute with an order entity that records an order for this product; only the product ID would be included. The relational theory would stipulate a join on the order table and product table to determine the product group and any other attributes of the specified product. This relational theory concept does not apply to a data warehousing system because in a data warehousing system one may be capturing the group that this product belonged to when the order was filled. Even though the product moves to different groups over time, the relationship between the product and the group in context of this particular order is static.

Example - 2

Another important example can be the price of a product. The prices in an operational system may change constantly. Some of these price changes may be carried to the data warehouse with a periodic snapshot of the product price table. In a data warehousing system one would carry the list price of the product when the order is placed with each order regardless of the selling price for this order. The list price of the product may change many times in a year and the product price database snapshot may even manage to capture all these prices. But, it is nearly impossible to determine the historical list price of the product at the time each order is generated if it is not carried to the data warehouse with the order.

The relational database theory makes it easy to maintain dynamic relationships between business entities, whereas a data warehouse system captures relationships between business entities at a given time. The concept of de-normalization and transformation is illustrated in the figure 1.1 below.

Related Topic

Data Normalization

The relational database theory was formulated in the late sixties by a researcher at IBM, E. F. Codd. Many prominent researchers have made significant contributions to this model since its introduction. Today, most of the popular database platforms follow this model closely. A relational database model is a collection of two-dimensional tables consisting of rows and columns. In the relational modeling terminology, the tables, rows, and columns are respectively called relations, attributes, and tuples. The name for relational database model is derived from the term relation for a table. The model further identifies unique keys for all tables and describes the relationship between tables.

Normalization is a relational database modeling process where the relations or tables are progressively decomposed into smaller relations to a point where all attributes in a relation are very tightly fixed with the primary key of the relation. Most data modelers try to achieve the “Third Normal Form” with all of the relations

before they de-normalize for performance or other reasons. The three levels of normalization are briefly described below:

i. First Normal Form

A relation is said to be in First Normal Form if it describes a single entity and it contains no arrays or repeating attributes. For example, an order table or relation with multiple line items would not be in First Normal Form because it would have repeating sets of attributes for each line item. The relational theory would call for separate tables for order and line items.

ii. Second Normal Form

A relation is said to be in Second Normal Form if in addition to the First Normal Form properties, all attributes are fully dependent on the primary key for the relation.

iii. Third Normal Form

A relation is in Third Normal Form if in addition to Second Normal Form, all non-key attributes are completely independent of each other.

The process of normalization generally breaks a table into many independent tables. While a fully normalized database can yield a flexible model, it generally makes the data model more complex and difficult to follow. The performance of a fully de-normalized database system would be very low. A data modeler in an operational system would take normalized logical data model and convert it into a physical data model that is significantly de-normalized. De-normalization reduces the need for database table joins in the queries.

The reasons for de-normalizing the data warehouse model are the same as they would be for an operational system, namely, performance and simplicity. The data normalization in relational databases provides considerable flexibility at the cost of the performance. This performance cost is sharply increased in a data warehousing system because the amount of data involved may be much larger. A

query with relatively small tables of an operational system may be acceptable in terms of performance cost, but the same may take unacceptably long time with large tables in the data warehouse system.

Figure 1.1 – Data Warehouse – Denormalization & Transformation

2. OPERATIONAL TERMS

The terms and names used in the operational systems are transformed into uniform standard terminology by the data warehouse transformation processes. The operational application may use cryptic or difficult to understand terms for a variety of different reasons. The platform software may impose length and format restriction on a term, or purchased application may be using a term that is too generic for the business. The data warehouse needs to consistently use standard business terms that are self-explanatory.

Example - 3

A customer identifier in the operational systems may be called cust, cust_id, or cust_no. Further, different operational applications may use different terms to refer

to the same attribute. For example, a home loan customer of a bank may be referred to as an HL_Borrower whereas a Personal Loan customer may be referred as PL_Borrower. One may choose a simple standard business term such as Customer Id in the data warehouse. This term would require little or no explanation even to a layman.

3. ATTRIBUTE DEFINITION

Different systems may evolve to use different lengths and data types for the same data element. One system may have the product ID to be either 12 or 14 numeric characters, whereas another system may accommodate product IDs of up to 18 alphanumeric characters. The software of an operational application may support very limited data types and it may impose severe limitations on the names. Software of another application may support a very rich set of data types, and it may be very flexible with the naming conventions. As an attribute is defined physically for the data warehouse, it is essential to use meaningful data types and lengths. Use the standard data length and data type for each attribute everywhere it is used. A functional data dictionary or a reference can facilitate this consistent use of physical attributes.

4. VALUE CONSISTENCY

All attributes in the data warehouse need to be consistent in the use of predefined values. Different source applications invariably use different attribute values to represent the same meaning. These different values need to be converted into a single, most sensible value as the data is loaded into the data warehouse.

Example - 4

A simple example for the consistent use of entity attributes is the use of a gender flag for an individual. One source application may use flags such as “M” and “F” to store gender for an individual whereas another application may use the detail “Male” and “Female” to store gender. Other applications may use yet other values to store the same piece of information. The data warehouse may choose to

consistently use “M” and “F” for gender for all individuals throughout the system. A more complex example can be the case of dealing with complex data values in the source application. Many older applications use single data value to represent multiple attributes. An account number, for example, may not only represent a unique account but also it may also represent the account type. All accounts starting with 4000 may represent one type of account whereas all other accounts may represent something else to the business. The data warehouse would consistently use the account ID to only represent a unique account. The account type may be computed and saved as a separate attribute.

5. PHYSICAL MODEL

The data warehouse model outlines the logical and physical structure of the data warehouse. As opposed to the archived data of the legacy systems, considerable effort needs to be devoted to the data warehouse modeling. This data modeling effort in the early phases of the data-warehousing project can result in the development of an efficient data warehouse that is expandable to accommodate all of the business data from multiple operational applications, thus providing significant benefits to the organization.

The data modeling process needs to structure the data in the data warehouse independent of the relational data model that may exist in any of the operational systems. The data warehouse model is likely to be less normalized than an operational system model. Further, the operational systems are likely to have large amounts of overlapping business reference data. Information about current products is likely to be used in varying forms in many of the operational systems. The data warehouse system needs to consolidate all of the reference data. For example, the operational order processing system may maintain the pricing and physical attributes of products whereas the R&D department may maintain design and formula attributes for the same product. The data warehouse reference table for products would consolidate and maintain all attributes associated with products that are relevant for the analysis processes. Some attributes that are essential to

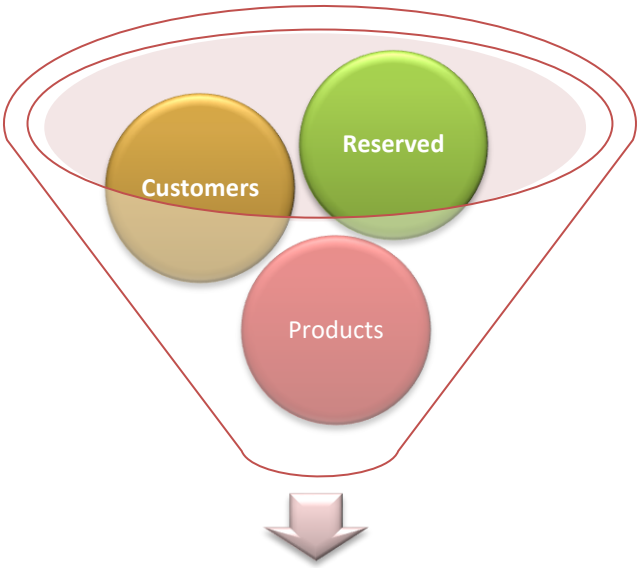
the operational system are likely to be deemed unnecessary for the data warehouse and may not be loaded and maintained in the data warehouse.

Key Points

- The data warehouse model needs to be extensible and structured such that the data from different applications can be added as a business case can be made for the data.
- A data warehouse project in most cases cannot include data from all possible applications right from the start.
- Most of the successful data warehousing projects take an incremental approach to adding data from the operational systems and aligning it with the existing data.
- They start with the objective of eventually adding most if not all business data to the data warehouse.
- Keeping this long-term objective in mind, they may begin with a couple operational applications that provide the most important data for business analysis.

The Figure 1.2 below illustrates the extensible architecture of the data warehouse.

Figure 1.2 – Components of a Data Warehouse



6. LOGICAL MODEL

A data warehouse logical model aligns with the business structure rather than the data model of any particular application. The entities defined and maintained in the data warehouse parallel the actual business entities such as customers, products, orders, and distributors. Different parts of an organization may have a very narrow view of a business entity such as a customer. For example, an housing loan department of a bank may only know about a customer in the context of the personal loan department. Another group in the same bank may know about the same customer in context of a recurring deposit account. The data warehouse view of the customer would transcend the view from a particular part of the business. A customer in the data warehouse would represent a bank customer that has any kind of business with the bank.

A data warehouse would most likely build attributes of a business entity by collecting data from multiple source applications. Consider, for example, the demographic data associated with a bank customer. The retail operational system may provide some attributes such as PAN number, address, and phone number. A loan system or some purchased database may provide with employment,

income, and net worth information. The structure of the data in any single source application is likely to be inadequate for the data warehouse. The structure in a single application may be influenced by many factors, including:

Third Party Applications

The application data structure may be dictated by an application that was purchased from a software vendor and integrated into the business. The user of the application may have very little or no control over the data model. Some vendor applications have a very generic data model that is designed to accommodate a large number and types of businesses.

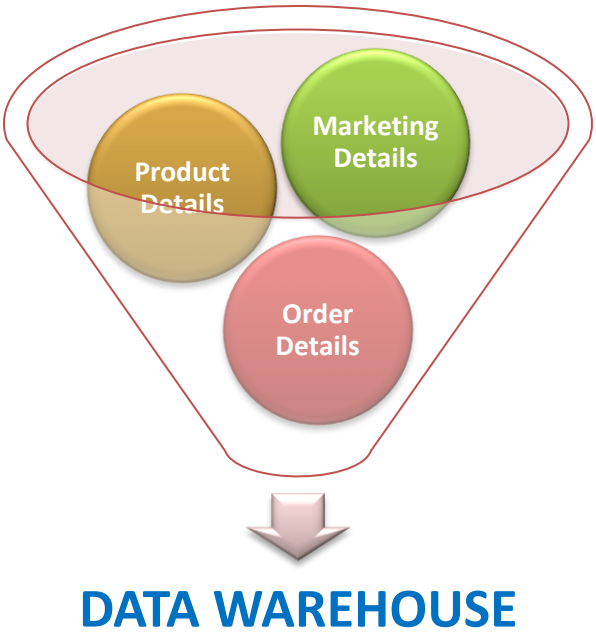
Legacy Applications

The source application may be a very old proprietary application where the data model has evolved over the years. The database engine in this application may have been changed more than once without anyone taking the time to fully exploit the features of the new engine. There are many legacy applications in existence today where the data model is neither well documented nor understood by anyone currently supporting the application.

Platform Limitations

The source application data model may be restricted by the limitations of the hardware/software platform or development tools and technologies. A database platform may not support certain logical relationship or there may be physical limitations on the data attributes. The following figure 1.3 depicts the alignment of the data warehouse entities with the business structure. The data warehouse model breaks away from the limitations of the source application data models and builds a flexible model that parallels the business structure. This extensible data model is easy to understand by the business analysts as well as the managers.

Figure 1.3 – Data Warehouse – Key Entities



7. DATA STORAGE

Data from most of the operational systems is archived after the data becomes inactive. For example, an order may become inactive after a set period from the fulfillment of the order, or a bank account may become inactive after it has been closed for a period of time. The primary reason for archiving the inactive data has been the performance of the operational system. Large amounts of inactive data mixed with operational live data can significantly degrade the performance of a transaction that is only processing the active data. Since the data warehouses are designed to be the archives for the operational data, the data is saved for a very long period. The cost of maintaining the data once it is loaded in the data warehouse is minimal. Most of the significant costs are incurred in data transfer and ensuring its consistency. Storing data for more than five years is very common for data warehousing systems. Normally one would start with storing the data for two or three years and then expand to five or more years once the affluence of business knowledge in the data warehouse is discovered. The falling prices of hardware have also encouraged the expansion of successful data warehousing projects.

The separation of operational data from the analysis data is the most fundamental data-warehousing concept. Not only is the data stored in a structured manner outside the operational system, businesses today are allocating considerable resources to build data warehouses at the same time that the operational applications are deployed. Rather than archiving data to a tape as an afterthought of implementing an operational system, data warehousing systems have become the primary interface for operational systems. The figure 1.4 below highlights the reasons for separation as discussed in this section.

8. DATA TRANSFORMATION

The data is logically transformed when it is brought to the data warehouse from the operational systems. The logical transformation of the data brought from the operational systems to the data warehouse may require considerable analysis and design effort. The architecture of the data warehouse and the associated data warehouse model greatly impacts the success of an organizational warehousing project. This section introduces some of the most fundamental concepts of relational database theory that do not fully apply to data warehousing systems. Even though most data warehouses are deployed on relational database platforms, some basic relational principles are knowingly modified when developing the logical and physical model of the data warehouses.

It is essential to understand the implications of not being able to maintain the state information of the operational system when the data is moved to the data warehouse. Many of the attributes of entities in the operational system are very dynamic and constantly modified. Many of these dynamic operational system attributes are not carried over to the data warehouse; others are static by the time they are moved to the data warehouse. A data warehouse generally does not contain information about entities that are dynamic and constantly going through state changes. The above concept is highlighted in the followiing example: An order tracking system that tracks the inventory to fill orders. An order may go through many different stages or states before it is fulfilled or goes to the “closed” status. Other order status may indicate that the order is ready to be serviced, is

being serviced ready to be shipped, etc. This order entity may go through many states that capture the status of the order and the business processes that have been applied to it. It is nearly impossible to carry forward all of attributes associated with these order states to the data warehousing system. The data warehousing system is most likely to have just one final snapshot of this order. As the order is ready to be moved into the data warehouse, the information may be gathered from multiple operational entities such as order and shipping to build the final data warehouse order entity.

Figure 1.4 – Data Warehouse - Contents

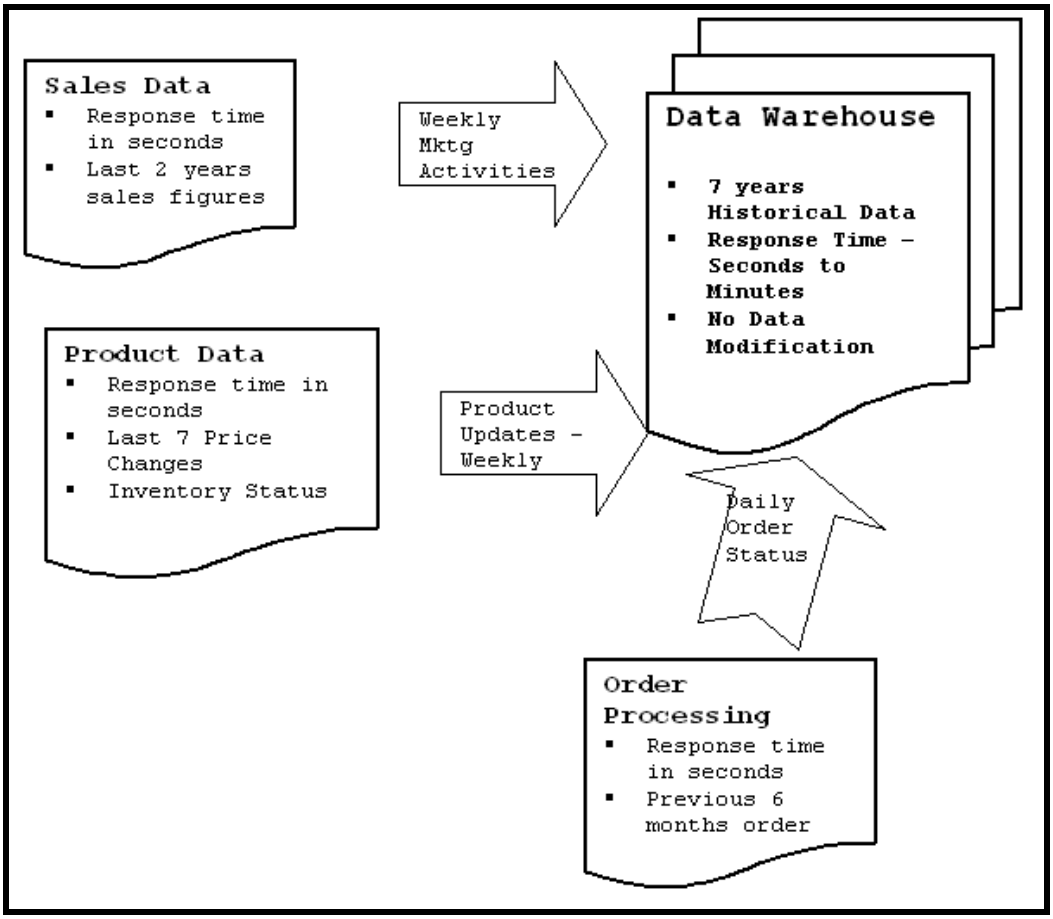
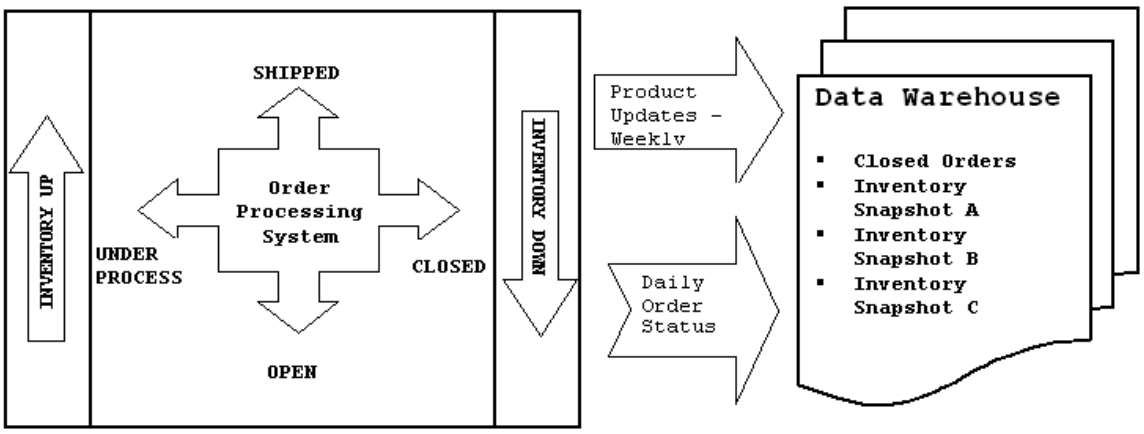


Figure 1.5 – Data Warehouse – Operational Information



The Figure 1.5 above illustrates how most of the operational state information cannot be carried over the data warehouse system. The inventory may change with every single transaction. The quantity of a product in the inventory may be reduced by a service order transaction or this quantity may be increased with receipt of a new product shipment. If this order processing system executes nine thousand transactions in a given day, it is likely that the actual inventory in the database will go through just as many states or snapshots during this day. It is impossible to capture this constant change in the database and carry it forward to the data warehouse. This is still one of the most perplexing problems with the data warehousing systems. There are many approaches to solving this problem. The most common method is to carry periodical snapshots of the inventory data to the data warehouse. This scenario can apply to a very large portion of the data in the operational systems. However this get much more complicated as extended time periods are considered.

Physical transformation of data results in its homogenization and cleansing. These data warehousing processes are typically known as “data scrubbing” or “data staging” processes. The “data scrubbing” process is one of the most time consuming and laborious processes within a warehousing project. However these processes cannot be eliminated, as it would result in the diminishing the analytical value of even the clean data. Physical transformation includes the use of simple or lucid standard business terms, and standard values for the data. A complete dictionary associated with the data warehouse can be a very useful tool. During

these physical transformation processes the data is sometimes “staged” before it is entered into the data warehouse. The data may be combined from multiple applications during this “staging” step or the integrity of the data may be checked during this process.

Note - 2

Historical data and the current operational application data are likely to have some missing or invalid values. It is important to note that it is essential to manage missing values or incomplete transformations while moving the data to the data warehousing system. The end user of the data warehouse must have a way to learn about any missing data and the default values used by the transformation processes.

9. DATA SUMMARIZATION

Most of the queries and reports that are build in data warehouse systems are simple aggregations based on predefined parameters. Another key attribute of the data warehouses is the predefined and automatically generated summary views. For example, many people in an organization may need to see sales figures for a particular product. They may have a need to summarize these sales figures for a week, a month, or a quarter. It may not be practical to summarize the needed data every time an analyst requires it. A data warehouse that contains summary views of the detail data around the most common queries can sharply reduce the amount of processing needed at the time of analysis. Summary views are typically created around business entities such as customers, products etc.

The summary views also hide the complexities of the detail data. Performance gain is the most significant tangible aspect of the summary views in the data warehouse. Most relational databases provide the ability to build views for users that hide the underlying tables. In most SQL server packages, including MS SQL Server, the view exists only as a definition and it is created at the time it is actually used. While the concept of summary views in data warehousing systems is similar, it important to not confuse data warehousing summary views with the term “views”

as it is used in a database system. A summary view in a data warehouse refers to an actual table that is created and maintained independent of when it used by a user.

10. VIEWS

Summary views often are generated not only by summarizing the detail data but also by applying business rules to the detail data. For example, the summary views may contain a filter that applies the exact business rules for considering an order of sale or a filter that applies the business rules for allocating a sale to a franchisee. The summary views can hide the complexities of the detail data from the end user.

The generation of summary view necessitates the application of complex business rules. These business rules may determine exactly what constitutes a sale or they may determine how a sale is allocated to a franchisee. Large organizations often have complex rules to charge sales to different franchisee accounts. Some sales may be allocated to warranty replacement and thus not be counted as sales. It is also possible that some sales may be further discounted based on a master contract with the customer and thus need to be reduced when calculating product sales for a period. A data warehouse will generally have more than one views based on business entities such as customers and products. There may be multiple physical tables or the same table may contain additional attributes that allow for easy queries.

In addition to applying the business rules while generating summary views, the data warehousing system may perform complex database operations such as multi-table joins. Product sales may be computed by joining the Sales, Invoice, and Product tables. The criteria to join these tables may be complex. While individuals mining data in the warehouse detail records need to understand all the complexities of business rules, most users can retrieve effective summary business information without fully understanding the detail data.

KEY ADVANTAGES

Performance Gain

The single most important reason for building the summary views is the significant performance gains they facilitate. Not only are all the complexities of detail data interpreted for an end user; the summary views also perform the most time-consuming data analysis before it is needed.

Summary views allow one to run a product sales query by merely setting up a filter based on indexed fields such as date, product codes, and other relevant criteria. The summary views will result in a query being run on a smaller table thus providing faster results and the usage of significantly reduced processing power. However in some instances a summary view table can be as large as the detail tables. This may be caused by summarization in very small units or combining multiple summary views into one data table. For example, one may not be able to summarize the product sales by week. Instead daily product sales figures may be required for some queries. Even in these large summary views, the performance is generally better because many of the table joins are eliminated and queries can generally use the indexes.

The summary views in a data warehouse provide multiple views into the same detail data. These views are predefined dimensions into the detail data. These views provide an efficient method for the analyst to link with the detail data when necessary. For example, for the sales order data, four different product sales summary views could be generated, summarizing weekly sales data. These views are summarized by product, customer, franchisee and state and include the same detail data that needs to be updated or regenerated as new data is brought into the data warehouse. Even though most of the analysis is likely to be done using the summary views, there needs to be a simple and robust way for an analyst to drill down into the detail data. Many business problems require review of the detail data to fully understand a pattern or anomaly exhibited in the summarized reports or queries. Drill down from many different summary views can lead to the same detail data. A single anomaly in detail data may manifest itself differently in different summary views.

Summarization and predefined analysis of data in a data warehouse system is an important task. It is essential to maintain the integrity of the summary views because very large parts of the data warehouse activity is against the summary views. It is important to note that the summary views needs to be maintained as new data comes into the data warehouse.

1.4 Key Design Issues

DESIGN ISSUES - SUMMARY

- Data Transformation
- Missing Values
- Operational System Integration
- Value Consistency
- Time Dimension
- Version Integration

The data brought into the data warehouse is sometimes incomplete or contains values that cannot be transformed properly. It is very important for the data warehouse transformation process to use intelligent default values for the missing or corrupt data. It is also important to devise a mechanism for users of the data warehouse to be aware of these default values. Some data attributes can easily be defaulted to a reasonable value when the original is missing or corrupt. Other values can be obtained by referencing other current data. For example, a missing product attribute such as unit-of-measure on an order entity can be obtained by accessing the current product database. Some attributes cannot be filled by defaults for missing values. In fact, it may be dangerous to attempt to assign default for certain types of missing values. A poor default may corrupt the data and lead to invalid analysis at a later stage. In these cases, it is safest to leave the missing values as blank. In some cases, it may make sense to pick a specific value or symbol that indicates a missing value. The timing of the start of the period for

which data is loaded into the data warehouse can be important. It is safest to load data in the data warehouse for complete years.

It is important to design a good system to log and identify data that is missing from the data warehouse. When a user runs a query against the data warehouse, it is essential to understand the population against which the query is run. Physical transformation of source application data requires considerable effort and it can be difficult at times, but a well-considered set of physical data transformations can make a data warehouse user-friendlier. Further, accurate and complete transformations help maintain the integrity of the data warehouse.

Data warehousing systems are most successful when data can be combined from more than one operational system. When the data needs to be brought together from more than one source application, it is natural that this integration be done at a place independent of the source applications. A data warehouse effectively combines data from multiple source applications such as sales, marketing, finance, and production. Many large data warehouse architectures allow for the source applications to be integrated into the data warehouse incrementally. The primary reason for combining data from multiple source applications is the ability to cross-reference data from these applications. All the date within a warehouse is built around the time dimension and is the primary filtering criterion for a very large percentage of all queries against the data warehouse. An analyst may generate queries for a given week, month, quarter, or a year. Another popular query in many data warehousing applications is the review of year-on-year activity. For example, one may compare sales for the first quarter of the year 2006 with the sales for first quarter of the years 2005 and 2004. The time dimension in the data warehouse also serves as a primary cross-referencing attribute. For example, an analyst may attempt to access the impact of a new marketing campaign run during selected months by reviewing the sales during similar periods. The ability to establish and understand the correlation between activities of different organizational SBU's is one of the primary features offered by data warehousing systems.

The data warehouse system can serve not only as an effective platform to merge data from multiple current applications but can also be used to integrate multiple versions of the same application. For example, an organization may have migrated to a new standard business application that replaces a legacy application. The data warehouse system can combine the data from the old and the new applications. A properly designed data warehouse can allow for continual analysis even though the base operational application has changed.

1.5 Data Warehouse Tools

A data warehouse is designed to be highly open and flexible. The data warehouse should be accessible by as many end-user tools and platforms as possible. However it may always not be possible to make every feature of the data warehouse available from all the end user tools employed within an organization.

PRIMARY TOOLS

- Standard Reports & Queries
- Summary Table Query
- Data Mining

The simple query capability built into most spreadsheets may be adequate for a user that only needs to quickly reference the data warehouse. Other users may require the use of the most powerful multi-dimensional analysis tools. The data warehouse administrators need to identify the tools that are supported for access to the data warehouse and the capabilities that are available using these different tools. In most data warehousing projects, there is a need to select a preferred data warehouse access tool for the most active users. A small number of users generate most of the analysis activity against the data warehouse. The data warehouse performance can be tuned to the requirements of the tool appropriate for these active users. This tool can be used for training and demonstration of the data warehouse. The following are some of the commonly employed warehouse access tools:

Standard Reports and Queries

Many users of the data warehouse need to access a set of standard reports and queries. It is desirable to periodically and automatically produce a set of standard reports that are required by many different users. When these users need a particular report, they can just view the report that has already been run by the data warehouse system rather than running it themselves. This facility can be particularly useful for reports that take a long time to run. However such facilities would require a client-server environment with the reports being accessed using the client program. This facility would need to work with or be part of the data warehouse access tool. Besides these an organization may also provide a web interface to the reports. In many data warehouse systems, this report and query server becomes an essential facility. The data warehouse users and administrators constantly need to consider any reports that are candidates to become standard reports for the data warehouse. Frequently, individual users may develop reports that can be used by other users. In addition to standard reports and queries, sometimes it is useful to share some of the advanced work done by other users. A user may produce advanced analysis that can be parameterized or otherwise adapted by other users in different parts of the same organization or even in multiple organizations.

Queries against Summary Tables

As introduced earlier, the summary views in the data warehouse can be the object of a large majority of analysis in a data warehouse. Most of the analytical activity within a warehouse is confined to simple filtering and summation from the summary views. These summary views contain predefined standard business analysis.

For example, in a typical data warehouse, the product summary view may account for a very large number of queries where different users select different products and the time periods for product sales and profit margin queries. These queries provide quick response and they are very simple to build. Advanced users typically

attach a pivot table in their analysis tool to data warehouse summary tables for simple multi-dimensional analysis.

Data Mining in the Detail Data

The data mining in the detail data accounts for a very small percentage of the data warehouse activity. However the most useful organizational data analyses are done on the detail data. The reports and queries off the summary tables are adequate to answer many "what" questions in the business. The drill down into the detail data provides answers to "why" and "how" questions.

Data mining is an evolving science. A data-mining user starts with summary data and drills down into the detail data looking for arguments to prove or disprove a hypothesis. The tools for data mining are evolving rapidly to satisfy the need to understand the behavior of business units such as customers and products.

1.6 Data Warehouse Interfaces

The data warehouse system would be interfaced with other applications that use it as the source of operational system data. A data warehouse may feed data to other data warehouses or smaller data warehouses called data marts. The operational system interfaces with the data warehouse are inherently stable. Since an organizational data warehouse is a reliable source of data that is consistently separate from the operational systems a single interface with the interfacing operational applications is much easier and more functional than multiple interfaces. The data warehouse can be a consistent source satisfying application needs for a variety of data as opposed to the operational systems. However it is important to note that much of the operational state information is moved onto a data warehouse. Hence a data warehouse cannot be a source for all operation system interfaces. Although a majority of the activity against most data warehouses is simple reporting and analysis, higher end complex analytical functions are being rapidly developed. The analysis performed by a warehouse is

much simpler and much cheaper from an organization standpoint and hence contributes significantly to its wide spread deployment.

1.7 Data Warehousing and Data Mining

A data-mining user starts with summary data and drills down into the detail data looking for arguments to prove or disprove a hypothesis. The tools for data mining are evolving rapidly to satisfy the need to understand the behavior of business units such as customers and products. Data Warehouses are intended to deliver information derived from a variety of operational systems to support the business analysis needs of an organization. While much of the challenge in building a successful data warehouse lies in its design and the transfer of data, an equally important challenge arises in the deployment of the data warehouse. Pre-defined queries and reports are typically used to satisfy routine information requirements. However in dynamic business environment users need answers to everyday business questions on an ad hoc basis. A handful of technically astute users are capable of serving themselves information with almost any tool, but reaching beyond the power user to the mainstream business user has proven to be a significant challenge. For this reason, many data warehouses have failed to meet their original goals. Deploying the data warehouse on the Internet, or on a private corporate intranet/extranet – makes the data warehouse available to anyone with a web browser. This eliminates the complexities involved in the installation and administration of data analysis (OLAP) tools on each client machine. Further it also eliminates the need to train and support users to operate all the complex OLAP tools. Targeting such a large user population requires a new and more natural way for users to interact with their computers. One of the principal reasons for developing a Data Warehouse is to integrate operational data from various sources into a single and consistent architecture that supports analysis and decision-making within the enterprise. Operational (legacy) systems create, update and delete production data that "feed" the Data Warehouse.

1.8 Recommended Reading

Books

1. Ponniah Paulraj, Data Modeling Fundamentals: A Practical Guide for IT Professionals, Wiley-Interscience, ISBN-13: 978-0471790495
2. Kimball Ralph & Ross Mary et.al, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd Edition), 2008, Wiley ISBN-13: 978-0470149775
3. Inmon Bill, Building The Data Warehouse, 4th Edition, Wiley, ISBN-13- 978-0764599446

URL

http://www.1keydata.com/datawarehousing/concepts.html

http://www.principlepartners.com/presentations/DataWarehouseConceptsAndArchitecture.pdf

http://en.wikipedia.org/wiki/Data_warehouse

http://www.users.qwest.net/~lauramh/resume/thorn.htm

http://www.agiledata.org/essays/dataNormalization.html

1.9 Referred Standards

Meta Data Coalition (MDC) Open Information Model

Object Management Group (OMG) Common Warehouse Metamodel (CWM)

1.10 Key Terms [Nomenclature]

Business Process Outsourcing (BPO)

Business Process Re-engineering (BPR)

Cleansing

Data Mining

Data Scrubbing

Data Staging

Decision Support Systems (DSS)

De-normalization

Executive Information Systems (EIS)

Homogenization

International Business Machines (IBM)

Knowledge Management Systems (KMS)

Multi-Dimensional Analysis

Multi-table joins

Network Operating System (NOS)

Normalization

On-line Analytical Processing (OLAP)

Permanent Account Number (PAN)

Personal computers (PC)

Relational database management systems (RDBMS)

Research & Development (R&D)

Small Business Units (SBU)

Structured Query Language (SQL)

Summary views

1.11 Summary

Data Warehousing is a science that will continue to evolve with time. This chapter introduced the fundamental concepts of data warehousing along with its need and benefits. The concepts introduced in this chapter provide an indication of the scope an application of a data warehousing systems. The technological advances in the computing arena (both hardware as well as software) will continue to greatly influence the capabilities that are built into data warehouses. Data warehousing systems have become a key component of the organizational IT architecture.

KEY BENEFITS

- Improved Decision Making
- Enhanced Customer Services
- Information Re-Engineering

IMPROVED DECISION MAKING

The concept of data warehousing evolved out of the organizational need for easy access to a structured store of quality data in order to buttress its decision-making capabilities. It is a universally recognized and accepted fact that knowledge is one of the key assets that can be leveraged by organizations to provide significant benefits, including competitive advantages in the current knowledge economy. Organizations generally have stockpiles of data but find it increasingly difficult to access, analyze them and assimilate the key learning's in their day-to-day functioning. This is because of the diversity of operational platforms as well as the enormity of the various formats involved. This problem is further compounded by the storage of the data in radically different file and database structures developed by different vendors. Thus organizations thus would need to develop and employ scores of software programs to handle the extraction; processing and consolidation of the data, to meet the needs of the various analytical tools deployed leading to time and cost overruns.

Data warehousing offers a superior approach to avoid the above-mentioned problem. Data warehousing implements processes to access heterogeneous data sources with the ability to clean, filter, and transform the data while providing a structured storage mechanism, that is easy to access, understand, and use. This data can subsequently be used for querying, reporting, and data analysis. An added advantage of deploying an organizational data warehousing environment is the reduction of staff and other allied resources required to support queries and reports against operational and production databases. This brings about significant reduction in cost besides eliminating the resource drain on production systems due to the execution of time intensive complex queries. The multi-tiered data structure employed by warehouse facilitates enterprise analysis ranging from detailed transactional queries to high-level summary information, with an increased flexibility and quality resulting in better organizational decision making.

ENHANCED CUSTOMER SERVICES

The Data Warehouse architecture helps organization foster better relationships with its customers both internal and external. This is due to resultant correlation of all customer data via a single Data Warehouse architecture. This is a very crucial aspect in any organization irrespective of its domain, size or structure, more so in service providing entities. For example a customer requiring financial services from a banking provider would not require submitting certain basic details and may be accorded additional facilities based on his past relationship with the business entity.

RE-ENGINEERING

Path breaking ideas for reengineering key organizational business or allied processes may often be attributed to insightful information obtained form unlimited analysis of enterprise information. The task of defining the data warehouse requirements would also lead to the establishment of better enterprise goals and measures. Knowing what information is important to an enterprise will provide direction and priority for reengineering efforts. A Data Warehouse that is based

upon enterprise-wide data requirements provides a cost-effective means of establishing both data standardization and operational system interoperability. Data Warehouse development can be an effective first step in reengineering or remodeling an organizations legacy systems.

1.13 Check Your Learning

Review Questions

1. The organizational information analysis requirements are met by:
    a. Databases
    b. Data Warehouses
    c. Decision Support Systems
    d. Operating Systems
2. Warehouse is a structured extensible environment that is periodically updated and maintained for a length of time.
    a. True
    b. False
3. Data Warehousing systems support very sophisticated online analysis including multi-dimensional analysis.
    a. True
    b. False

4. Operational systems are designed for acceptable performance for _____transactions.
    a. On-the-fly
    b. Pre-defined
    c. All
    d. Specific
5. A key attribute of the data within a data warehouse system is that it is loaded on to the warehouse after it has become _____.

a. Non Volatile
b. Obsolete
c. Operational
d. Redundant

6. Normalization is a warehouse modeling process where the relations or tables are progressively decomposed into smaller relations to a point where all attributes in a relation are very tightly fixed with the primary key of the relation.
    a. True
    b. False
7. The data is logically _____when it is brought to the data warehouse from the operational systems.
    a. Complete
    b. Incomplete
    c. Transformed
    d. Appended
8. The process of combining data from multiple applications before being moved into a Data Warehouse is referred to as:
    a. Transformation
    b. Combining
    c. Staging
    d. Warehousing

9. The time dimension in the data warehouse serves as a primary cross-referencing attribute.
    a. True
    b. False
10. Data Warehouses are intended to deliver information derived from a variety of operational systems to support the _____ needs of an organization.

a. Archival
b. Storage
c. Business Analysis
d. None of the Above

Exercises

1. What is a Data Warehousing? Explain the organizational need and benefits for deploying data warehousing systems.
2. List down three major driving forces responsible for the widespread growth of data warehousing systems with a brief explanation for each.
3. List down five key attributes of a data warehouse along with a brief explanation.
4. What do you understand by the term “summary views”? List down the advantages of employing summary views.
5. Enumerate on the commonly employed organizational data warehousing tools.

Research Activities

Taking an example of a small organization or an SBU list down the following:

1. Needs and Benefits of implementing warehousing solutions
2. Database Design
3. Interfaces required
4. Access tools required

Chapter Objectives

This chapter lays the foundation for preparing the blue print for designing a data warehouse. It provides an introduction to the commonly used data analysis techniques and their linkage with the data warehouse architecture. The choice of the technique employed is based on the end user requirements and greatly impacts the modeling of the warehouse. The chapter begins with an introduction to the commonly employed data analysis tools and proceeds to list down the key warehouse components. The various structural options of a data warehouse are also subsequently introduced. Data Warehouse modeling is an important area that requires specific treatise and a model is constructed directly from the Enterprise Data Model, which is the high-level data blueprint describing the organization's integrated information requirements. This ensures that the collective information requirements of the enterprise are represented in the Data Warehouse Model and further each subject area in the Enterprise Data Model has a corresponding Data Warehouse Model component.

The emerging knowledge based economy has spurned organizations to deploy warehousing solutions with the hope of instantly exploiting them to provide tailor made solutions to strategic and operational issues. Thus a small note on multidimensional analysis has been included in this chapter. Multidimensional analysis has become a popular way to extend the capabilities of query and reporting. It provides an alternative to the cumbersome method of submitting multiple queries and has its data suitably structured to enable fast and easy answers to the questions typically raised.

The chapter also presents a brief on the Data warehouse Engineering Life Cycle. This would include the enterprise needs identification which is crucial component of the lifecycle. The design & development of operational systems necessitates a clear understanding of the requirements. Of these a lucid perception of the system look, feel and function is very essential. Data Warehouse implementation includes loading the preliminary data, implementing transformation programs, designing an

optimal user interface as well as querying and reporting mechanisms and finally supporting the effort by training the end users. Thus this chapter covers the essential knowledge required to design, implement and deploy data warehousing solutions tailor made to suit individual/organizational requirements

KEY LEARNING'S

- Data Analysis Techniques
- Data Models & Modeling Techniques
- Data Warehouse – Structure & Composition
- Data Warehouse Architecture
- Data Warehouse Engineering – Life Cycle

Chapter 2

DATA MODELING & DATA ANALYSIS

2.1 Introduction

A data warehouse is designed to provide easy access to high quality data sources. An important thing to note is that a warehouse is not the end objective, but rather the path leading to the end point. The end point is subsequent application of analytical and decision making tools to garner valuable insights from the extracted data. There are several techniques for data analysis that are in commonly employed. These include standard query and reporting and the more advanced multidimensional analysis and data mining techniques and is as illustrated in the figure 2.1 below. These techniques facilitate the formulation and display of query results, multi perspective analysis of data content, pattern discovery while clustering of attributes in the data provides valuable insights to an individual and/or an organization. There are several commonly employed methods of data analysis. The choice of these methods can greatly impact the type of data model selected and its content. For example for an organization requiring quick information retrieval (Query & Reporting capability) would deploy a model that structures the data in a normalized fashion. Query and reporting capability primarily consists of selecting associated data elements, summarizing them and grouping them by category, and presenting the results using direct table scans. For this type of capability a model with a normalized and/or denormalized data structure would be highly appropriate. Similarly a dimensional data model would be more appropriate if the objective is to perform multidimensional data analysis. This type of analysis requires that the data model support a structure that enables fast and easy access to the data on the basis of any of numerous combinations of analysis dimensions.

Multidimensional analysis requires a data model that facilitates easy access and presents multi dimensional viewing perspectives. The presence of a number of dimensions warrants the need for quick access to the data. If a highly normalized data structure were to be used, many joins would be required between the tables holding the multi dimensional data, thereby significantly degrading system performance. The following example throws further light on this concept. A sales executive needs to know the quantity of a specific product sold on a specific day, in a specific super market, in a specific price range. Then for further analysis he might also need to find the number of super markets selling the specific product, in a specific price range, on the specific day. These two queries require similar information, however one is viewed from a product perspective and the other viewed from the super market perspective. In this case, a dimensional data model would be most appropriate. An understanding of the data and its use will impact the choice of a data model. However in practice it is observed that most organizational implementations employ multiple types of data models to best satisfy the varying requirements of the data warehouse.

2.2 Data Analysis Techniques

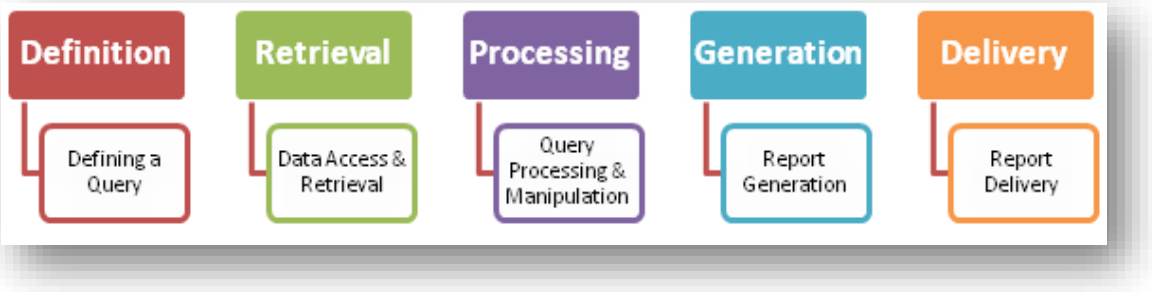
This section provides an introduction to the commonly used data analysis techniques and their linkage with the data warehouse architecture. The choice of the technique employed is based on the end user requirements and greatly impacts the modeling of the warehouse.

The query and reporting tool is one of the most commonly employed data analysis technique. Query and reporting analysis is the process of posing a question to be answered, retrieving relevant data from the data warehouse, transforming it into the appropriate context, and displaying it in a readable format. It is primarily driven by analysts who must pose those questions to receive an answer. This process is however considerably different from data mining, which is data driven. Traditionally most queries are two-dimensional or in other words have the capability of handling only two factors simultaneously. The standard sales or marketing queries

regarding daily or weekly or monthly product sales figure is an appropriate example of a two-dimensional query. Subsequent queries would then be posed to perhaps determine the quantity of a product was sold by a particular super market chain. The process flow in a standard query and reporting process is aptly illustrated in the following figure 2.2.

Query definition is the process of taking a business question or hypothesis and translating it into a query format that can be used by a particular decision support tool. When the query is executed, the tool generates the appropriate language commands to access and retrieve the requested data, which is returned in a format referred to as an answer set. The data analyst then performs the required calculations and manipulations on the answer set to achieve the desired results. Those results are then formatted to fit into a display or report template that has been selected for ease of understanding by the end user. This template could consist of combinations of text, graphic images, video, and audio. The report is finally delivered to the end user on a desired output media, which could be a hard copy, soft copy or a visual or audio display. The process of query and reporting thus commences with a query definition and ends with a report delivery.

Figure 2.1 - Query & Reporting Process



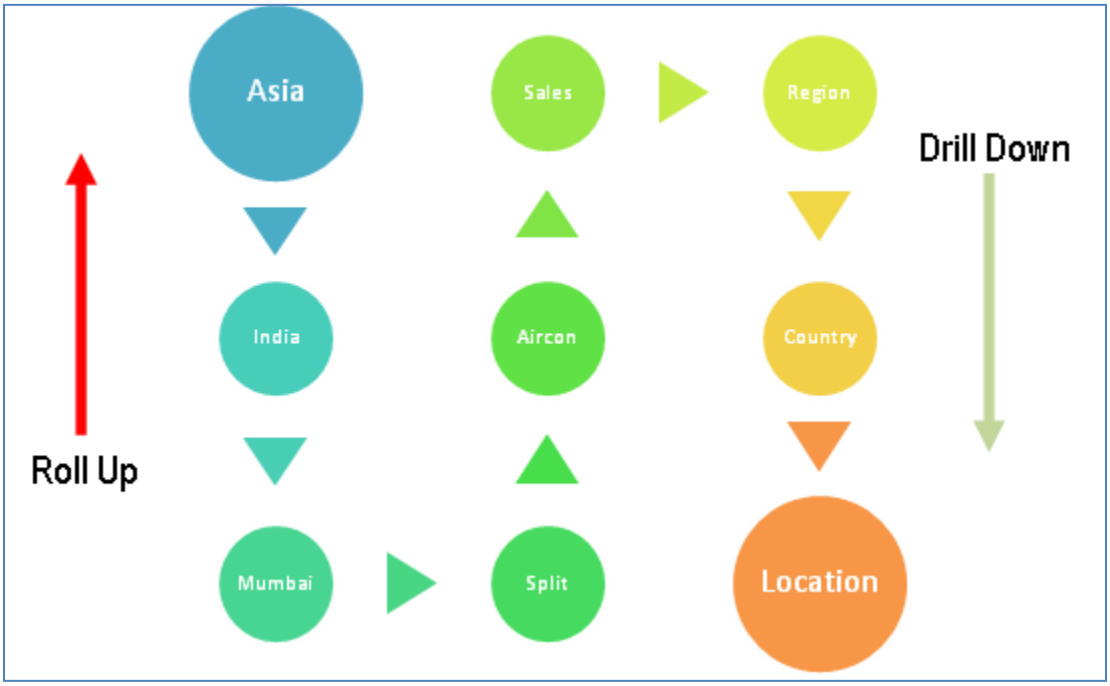
Multidimensional analysis is a method that extends the capabilities of query and reporting tools. This technique employs the use of structured data to enable fast and easy access to answers to typical questions as opposed to submitting multiple queries. For example, the data would be structured to include answers to a question related to the sale of a particular product on a particular day by a particular salesman for a particular super market. Each separate part of that query is

referred to as a dimension. The answers to each sub query is pre-computed, within the broad framework, By pre-calculating answers to each sub- query within the larger context, multiple answers can be made readily available. The results are not recalculated with each query but are simply accessed and displayed. The result of the query mentioned in the earlier example would automatically answer the following sub-query: The quantity of a particular product sold by a particular salesperson.

End users especially those with business orientation find the dimensional data (data categorized by different factors) easier to assimilate and disseminate. Dimensions can have individual entities or a hierarchy of entities, such as region, store, and department. Multidimensional analysis enables users to view a large number of interdependent factors involved in a business problem and to view the data in complex relationships. Typically end users are interested in exploring the data at different levels of detail and this requirement is generally dynamic in nature. The complex relationships are analyzed through an iterative process that includes drilling down to lower levels of detail or rolling up to higher levels of summarization and aggregation. A user can start by viewing the total sales for an organization and drill down to view the sales by continent, region, country, and finally by customer. Or, the user could start at customer and roll up through the different levels to finally reach total sales.

Pivoting or changing the dimension of the data can also be employed. Pivoting is a data analysis operation whereby a user takes a different viewpoint that is typical of the results of the analysis, changing the way the dimensions are arranged in the result. Like query and reporting, multidimensional analysis continues until no more drilling down or rolling up is performed. The end users have the option of performing drill down or roll up when using multidimensional analysis.

Figure 2.2 - Multi Dimensional Analysis

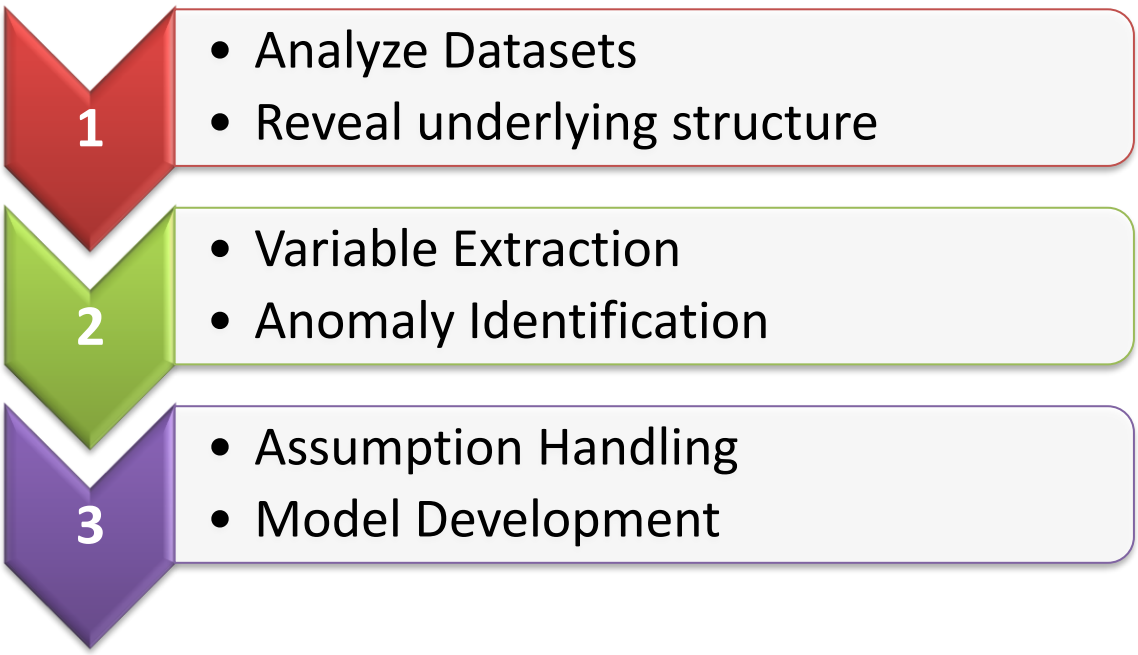


Data mining is a data analysis technique, which is very different from query and reporting well as multidimensional analysis. Using the techniques listed in the above section, a user has to create and execute queries based on hypotheses. Data mining searches for answers to questions that may have not been previously asked. This discovery could take the form of finding significance in relationships between certain data elements. This involves a clustering of specific data elements or other patterns governing the usage of specific sets of data elements. The discovery of these patterns can be followed by the deployment of algorithms to infer rules. These rules can then subsequently be used to generate a model that can predict a desired behavior, identify relationships among the data as well as discover patterns and group clusters of records with similar attributes.

Data mining is employed for statistical data analysis and knowledge discovery. The statistical data analysis detects unusual data patterns and applies statistical and mathematical modeling techniques to infer the patterns. The models are then used to forecast and predict. Types of statistical data analysis techniques include:

Knowledge discovery extracts implicit, previously unknown information from the data. This often results in uncovering unknown business facts. Data mining is data driven and unravels a high level of complexity in stored data and data interrelations in the data warehouse that is normally difficult to discover. It offers new insights into the business that may not be discovered with query and reporting or multidimensional analysis and provides users with answers to questions that have never been thought of.

Exploratory Data Analysis (EDA) is data analysis approach that employs a variety of techniques predominantly graphical techniques to:



EDA is an approach that is not identical to statistical graphics but lays emphasis on how data analysis should be carried out. The two terms, EDA and statistical analysis are however used almost interchangeably. Statistical graphics is a collection of graphic techniques that focuses on a single aspect of data characterization. EDA, in comparison, employs a direct approach wherein the data is used to reveal its underlying structure rather than making assumptions of the underlying data model. As opposed to a collection of techniques, EDA presents a methodology that facilitates an analysis of the objects/values to aid identification and interpretation. This is achieved by deploying a compilation of techniques that

are graphical in nature with a few quantitative techniques. The need for graphical interpretation is fuelled by the desire to facilitate and open ended exploration of the data set including its structural framework while revealing new and often hidden insights into the data. In combination with the natural pattern-recognition capabilities graphics provides a powerful and simple method for data analysis.

2.3 Data Models & Modeling Techniques

In order to analyze the behavior of business processes it is necessary to process a set of numeric values like sales revenue and shipment quantities. There may be also requirements for calculating or scrutinizing quality measures such as customer satisfaction rates, delays in the business processes and delayed or incorrect shipments. There may also be a need for analysis of the effects of business events or transactions with a view of extrapolating predictions for the future. The data displayed may cause the user to formulate another query to clarify the answer or gather more detailed information. This process continues until the desired results are reached. The type of analysis that will be done with the data warehouse can determine the type of model and its contents. Since query, reporting and multidimensional analysis require summarization and explicit metadata, it is important that the model contain these elements. Further multidimensional analysis usually entails drilling down and rolling up, so these characteristics need to be present in the model as well. A comprehensible data warehouse model is a precursor to ensuring consistency of results while retaining simplicity of end user tasks. An important point to note is that the deployment of data mining techniques warrants a model that provides for the lowest level of detail.

The graphical representation of the data for an organizational business domain or area of operation is referred to as a data model. The scope could include the entire organizational data needs – Enterprise Data Model That area of interest may be as broad as all the integrated data requirements of a complete business organization (Enterprise Data Model) or as focused as a single business area (SBU) or application. The data model represents the organizations functional area

within the business area (sales in telecom, IT, etc) or a specific domain being analyzed (product delivery, customer satisfaction). The important or recommended characteristics of a good data model are as listed below:

- *Entities (tables)*
- *Attributes (columns)*
- *Data inter-relationships*
- *Data cardinality, business rules governing data relationships*
- *Entities & attributes definition*
- *Well defined Primary and Secondary keys*
- *Graphical*

The data model communicates the meaning of the underlying data along with their attributes, inert-relationships and accurate definitions. A data model is the standard and accepted way of analyzing data, designing and implementing databases. The organizational data model does not change drastically over a length of time unless there is a fundamental change in the organization vision/mission. However the data usage and the processes involved can greatly vary across organizations in the same industry. This is despite the fact that their data requirements can be very similar. This similarity of data facilitates the development of template data models that can be adopted by organizations operating in a similar domain. From the above discussion one can infer that the organizational data models are stable while the process models may be volatile. The CASE[1][2] tools are commonly employed for data modeling. CASE tools provide the supporting software for development of the model including the graphics, data dictionary, links to other tools and supporting utilities. The function of the data model is to clearly convey data, data relationships, data attributes, and data definitions along with the business rules that governing the data. Data models are the accepted way of representing and designing databases.

[1] Computer-aided software engineering (CASE) tool refers to the holistic software used for automated system software development including design, analysis and programming. The CASE tool provides an automated environment for designing, developing and documenting structured computer programs. It also includes data modeling tools.
2 http://en.wikipedia.org/wiki/Computer-aided_software_engineering

1. TEMPLATE DATA MODELS

Template data models are fully functional pre-designed data models built for a specific industry. These models closely approximate the results achieved from the development of tailor made models for individual organizations. Template data models can be built for every conceivable data-modeling requirement, including the following types of applications:

Further the template data models of share the following characteristics:

- *Constructed for a specific industry or industry segment*
- *Clear, unambiguous, detailed and fully attributed*

Template data models are based upon detailed industry analysis that enables fully attributed models to be developed. This attention to detail is what accelerates the planning, analysis and design phase and makes the use of template data models of real value. Each entity should closely approximate a table that a Data Analyst would use to design an application and a Data Base Administrator would use to build that application. Every entity and attribute must be completely defined in conjunction with the organizational requirement and should include appropriate examples. It is not unusual for a suite of industry template data models to be supported by high volumes of documentation. Also adequate care should also be taken to see that every relationship is taken into account and properly named.

Key Features & Advantages

- The typical information technology (IT) project consists of a long planning, analysis, design and implementation phase incorporating a host of hardware, software and staffing activities. The most challenging, expensive and difficult to predict are the planning, analysis and design project phases

- Research has uncovered that over 60% of data warehouse implementations failures happen in the initial phases. These may not necessarily be as a result of fallacy in the hardware or software selection process. It is important to note that decisions on the data content, its

structure and representation with the data warehouse are made during the initial phase

- There is no method available to shorten the amount of time required for planning, analysis and design. There is no procedure or method to predict the result or the value (ROI) of these initial phases unless a set of detailed, industry-specific template data models are used to bootstrap these activities
- Template data models provide a close approximation of what would be achieved during that lengthy period at a small fraction of the cost.
- Template data models are the deliverables that is intended to kick-start and dramatically shorten the planning, analysis and design phase.
- The deployment of these models can save the project months of work while effecting massive monetary savings as well.
- Template data models are flexible and are designed to be modified, extended and integrated with other data models
- The commonly available template data models include the following:
    a. *Resource Estimation*
    b. *Gap Analysis*
    c. *Industry Knowledge Transfer*
    d. *Project Planning*
    e. *ROI Estimation*
    f. *Standards Definition*
    g. *Training*
- Building data models requires a unique set of skills which includes the following:
    a. *Domain Specific Industry Experience*
    b. *Data Modeling Expertise*
    c. *DBA Expertise*

d. *System Integration Exposure*

- The above combination of skills, expertise and long development cycles makes template data models unique. It also makes them extremely valuable in almost all industrial areas including:

    a. *Banking*
    b. *Financial Services*
    c. *Hospitality*
    d. *Insurance*
    e. *Pharmaceutical*
    f. *FMCG*
    g. *Retail*
    h. *Semiconductor & Allied Industries*

Models need to be frequently combined or integrated with other models to satisfy organizational requirements. The data models are developed from a common core of building blocks so they can be rapidly integrated with organizational data structures while providing consistent definitions of customer, distribution channels, geography and other common parameters. The data warehouse, data mart and allied applications must be mapped to the larger organizational context if it is to be successfully integrated with other systems. The larger organizational context is referred to as the 'Enterprise Data Model'.

2. ENTERPRISE DATA MODELS

The Enterprise Data Model is primarily employed for strategic planning as well as disseminating the warehouse data requirements throughout the enterprise. It is also used for implementing integrated systems while organizing data in the data warehouse along with the associated structures and applications. The Enterprise Data Model identifies the complete organizational data and is based upon in-depth analysis of business areas, terminology, data relationships, definitions, examples and business rules pertaining to specific industrial domains.

The Enterprise Data Model graphically depicts entities by subject area, keys, attributes, relationships and cardinalities. The Enterprise Data Model should be concise, easy to understand, supported by structured project information as well as definitions and presented graphically. Such a model would provide a point of integration for the entire organization while serving as a powerful tool for understanding the business and planning to make it more efficient and effective. The Enterprise Data Model is specific to an industrial environment or domain and provides the integrated primary data requirements of a standard organization operating in the domain. The model contains 4-10 entities representing the business subject area (domain), business functional area, and the key relationships between primary data. A typical Enterprise Data Model would consist of over 400 entities and 2,500 attributes.

3. BUSINESS DATA MODELS

Business Area Models are detailed data models representing standard functions within a specific business domain. The model is developed from a set of core of entities drawn from the higher-level subject area of domain specific Enterprise Data Models. The subject area model is subsequently expanded in scope and detail until its functionality is sufficient to support Decision Support Systems (DSS) or application development related to the functional business area under consideration. The core subject area entities from the Enterprise Data Model form the nucleus upon which detailed Business Area Models are developed. The advantage of developing subject area models directly from the Enterprise Data Model is that it insures that the keys will match and supports the future integration of data with other subject area data models, the Data Warehouse Model or back into the Enterprise Data Model itself. The following are examples of Business Area data models:

a. *Asset Management*
b. *Bookings & Billings*
c. *Budgeting*
d. *Channel Management*

e. *Commissions Management*
f. *Contract Management*
g. *Customer*
h. *Customer Sales Management*
i. *Financial*
j. *Forecast*
k. *Geography*
l. *HR/Employee*
m. *Inventory*
n. *Manufacturing/Shop Floor Control*
o. *Market*
p. *Marketing Events*
q. *Order*
r. *Pricing*
s. *Problem Reporting*
t. *Product Management*
u. *Prospective Customer Management*
v. *Purchasing*
w. *Training & Education*

Business Area Models contain the greatest level of detail and represent the low-level details of the data in the hierarchy. The new information is learned about business areas are subsequently added to the corresponding Business Area Model. This information may also be incorporated into the Enterprise Data Model, Data Warehouse Model or application data models. The usage of Business Area Models as a basis for analysis and design provides a solid foundation of industry-specific knowledge that leads to accelerated planning and development. Individual models can easily be combined or integrated to create other models. For example, a "Customer Feedback" prototype data model may be quickly built from individual Business Area Models. Once the Enterprise Data Model and Business Area Model components are in place, it is possible to introduce the Data Warehouse Models. The Data Warehouse Model represents the integrated decision support and

information reporting requirements of the business. The Data Warehouse is the center of the decision support and reporting data architecture and represents the ultimate source of clean, consistent data for the entire organization. The Data Warehouse may be surrounded by any number of functional decision support systems or "data marts" serving the associated functional business areas. As data moves from the Data Warehouse to local DSS or "data mart" systems, control of the data is turned over to local administrators. The Data Warehouse remains the consistent source of reliable data.

4. SUBJECT AREA MODELS

Subject Area Models describe functional subject areas that are unique to an organization. Subject Area Models represent the lowest levels of data and provide the design foundation for the data warehouse, data marts, applications development, business analysis and strategic planning. Each Subject Area Model is constructed from a set of core entities drawn from related subject areas in the Enterprise Model. This ensures that Subject Area Models will have common keys, attributes and definitions throughout the enterprise data architecture. This approach also supports integration of existing models and development of new models. The new information learned subsequently is added to the Subject Area Model. It may also be incorporated into the Enterprise Model, Data Warehouse Model or application models. Subject Area Models provides a solid foundation of knowledge for development industry-specific application and information solutions.

2.4 Data Modeling – Key Steps

The Data Warehouse Model represents the actual organization of data within the warehouse. It describes the data structures and their inter-relationships. The domain specific organizational informational requirements are represented by the data model. A domain (business area of an organization) data warehouse model is directly derived from the Enterprise Data Model specific to the operational domain. The Enterprise Data Model logical data structures are the foundation for development of corresponding data warehouse data structures. The Data

Warehouse data structures model the organizational data structure. Following is a list of characteristics of a data warehouse model:

- *Stable data over a length of time*
- *Summarized data for DSS, clean and reliable data for data marts*
- *Integrated data from multiple sources*
- *Design driven by evolving information needs*
- *Business area, function or subject orientation*
- *Integrated organizational information access*
- *Granularity of data suitable for analysis over extended time periods*
- *Multiple levels of summarization*
- *Iterative construction grouped by subject area*

The Data Warehouse is the ultimate source of clean, consistent data for the entire organization. It forms the backbone of the DSS and allied analysis systems. The Data Warehouse may be surrounded by any number of functional DSS or "data marts" serving the associated functional business area. As data moves from the Data Warehouse to local DSS or "data mart" systems, control of the data is turned over to local administrators. The Data Warehouse remains a consistent source of data over time for the business organization and is independent of the local data processing. Data warehouse deign commences with the analysis of the organizational core business areas that would be the major contributory data source to the warehouse. Business Area data Models describe lower levels of detailed data appropriate to building applications and DSS/Data Marts for a explicit business domains. These models are constructed from a set of core of entities derived from subject areas in the Enterprise Data Model. This ensures that Business Area Models will be based on common key entities, have common keys, attributes and definitions through the data architecture. This approach also supports consistent integration of existing data models and development of new data models. Each subject area in the Enterprise Data Model has a corresponding Data Warehouse Model component. A Data Warehouse Model can be implemented in two distinct functional levels:

a. *Decision Support Data*
b. *Summarized Data*

The Level 1 DSS data model describes data at the lowest level of detail appropriate for detailed analysis and decision-making as illustrated in the following examples:

- *Orders from specific customers for specific time periods*
- *Channel wise sales for specific time period*
- *Sales revenue for specific products from specific channels for a specified period*

The Summary model depicts summarized data defined in the DSS model. The top management of an enterprise generally employs the summary data generated for strategic decision making. Some of the questions posed at this level include the following:

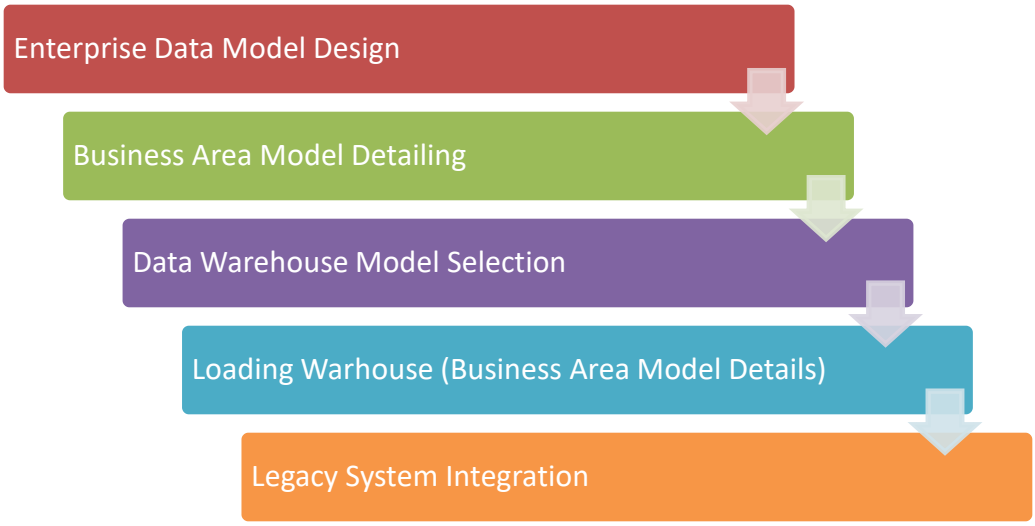
- *Total Revenues – Sales across products and customers during the financial year*
- *Total Orders across all channels for a time period*
- *Total Sales (in numbers/quantity) segmented channel wise for a specific period*

The listing down of the key organizational decision making factors is the precursor to building a template for the Data Warehouse Model. These factors may include following:

- *Customer Segments*
- *Markets and Market Segments*
- *Geography Definition*
- *Product Families*

The following figures 2.3 illustrate the key steps in constructing a data warehouse:

Figure 2.3 – Building a Data Warehouse



The Data Warehouse is built from existing template data model components representing the way the business intends to do business, which are then modified to meet the realities of legacy data and existing applications. The process is greatly accelerated by utilizes standard industry data building blocks that can readily be modified, extended or integrated to meet specific data requirements. Each of these models can be modified and contributes data to related models. The models are broken up into functional building blocks thereby facilitates parallel processing in multiple areas. Product data can be designed without waiting for the final Customer data structures. The foreign key relationships are defined immediately while the channel values can be defined later. The level of detail in the Business Area Models is consistent with that of the DSS/data marts. This makes it relatively easy to develop DSS/data marts that dovetail with the Data Warehouse. It also makes it simple to promote data structures from the Business Area Models into the Data Warehouse Model or related Business Area Models.

2.5 Data Warehouse – Structure & Composition

A data warehouse forms the primary repository of an organization's historical data or acts as the corporate memory of the organization. A data warehouse is optimized for being integrated with Online Analytical Processing (OLAP) systems.

KEY WORDS – *Subject Oriented, Time Variant, Non-Volatile, Integrated*

The warehouse contains the data which is used by the OLAP systems or the enterprise Decision Support Systems (DSS). The data warehouse contains the raw material for the organizational DSS. The data warehouse is optimized for reporting and analysis (OLAP) in comparison to operational systems that are optimized for simplicity and speed of modification (Online Transaction Processing or OLTP). This is achieved through the heavily normalized databases and an entity-relationship model. The data in Data Warehouses are heavily denormalized, summarized and/or stored in a dimension-based model in order to achieve acceptable query response times. Following are some of the important characteristics of a data warehouse:

1. **Subject-Oriented**

   The data in the database is organized so that all the data elements relating to the same real-world event or object are linked together

2. **Time-variant**

   The changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time

3. **Non-volatile**

   The data in the database is never over-written or deleted, but retained for future reporting

4. **Integrated**

   The database contains consistent data from most or all of an organization's operational applications.

2.5.1 Data Warehouse Structure

We will initially have a look at the structure of a data warehouse before proceeding to understand its key components. A data warehouse consists of two major parts as outlined below:

Physical Store

The physical store is a server based database that is employed for querying and includes an OLAP database for running reports. The physical store for the Data Warehouse includes one database that is employed for running the SQL queries. The physical store contains all the data that has been imported from different sources.

Logical Schema

The logical schema is the conceptual model that maps onto the data in the physical store. The logical schema provides an understandable view of the data in the data warehouse, and supports an efficient import process. For example, a developer can use the logical schema to modify the location of data stored in the underlying physical tables. A developer interacts with the logical schema in order to add, update, or delete data in the data warehouse. The end user need not be aware of the logical schema. A logical schema includes the following:

Class

Class refers to a logical collection of data members.

For example, the Administrator_User class contains data members with administrative system privileges.

Data member

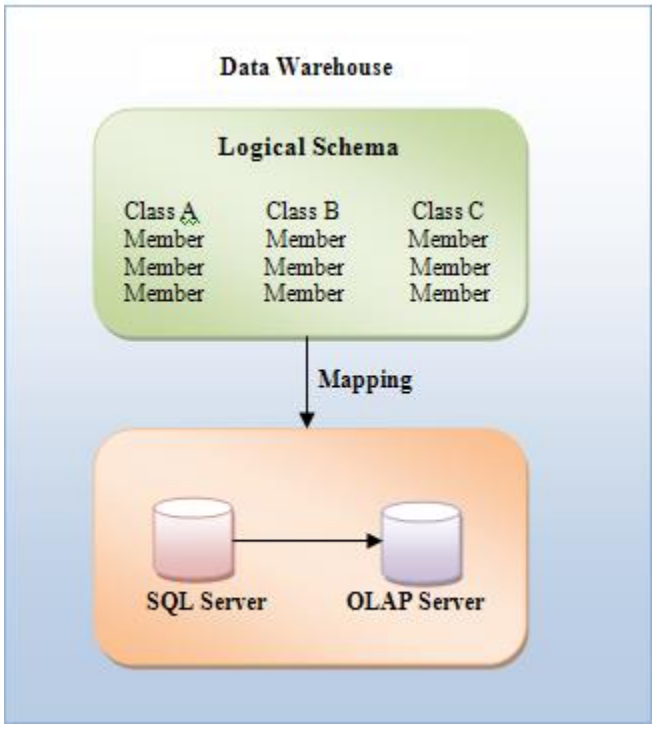
The data member is a structure that stores a piece of data. For example, the Employee_Number of the data member of the Administrator_User class stores the contact numbers for users with administrative system privileges.

Relation

A relation is a connection between two classes depicting a parent-child relationship. This relationship defines the number of instances of each class, and it provides the mechanism for sharing data members between classes. For example, Administrator_User is a parent to the child class Request. There can be many requests for one authorized user.

The logical schema uses classes, data members, relations, and other data structures to map data in the physical store. The interrelationship between the two entities is illustrated in the following figure 2.4below:

Figure 2.4 - Data Warehouse – Logical Schema



2.5.2 Data Warehouse Key Components

The primary components of a data warehouse are illustrated in the figure and described in more detail in the following section:

1. Data Sources

A data source refers to any electronic repository of information that contains relevant organizational data. This would include systems employed for OLTP as well as storage of operational data like main frames, mini frames, relational data base systems (Oracle, Microsoft SQL, IBM DB2), PC based databases like Microsoft Access, Microsoft Excel and any other electronic store of data. Data needs to be passed from these systems to the data warehouse either on a transaction-by-transaction basis for real-time data warehouses or on a regular cycle (e.g. daily or weekly) for offline data warehouses. Further organizational data embodied in physical documents would need to be digitized before porting onto the warehouse.

2. Data Transformation Layer

The data transformation layer is the subsystem responsible for the extraction of data from the data sources (source systems), transformation from the source format and structure into the target (data warehouse) format and structure and subsequently loading the transformed data into the data warehouse. Alternately another technique that is widely employed is to relegate the transformation of the source format and structure to the last stage of the transformation layer. Under this approach, the data is first extracted from the sources, loaded into the target data warehouse and then transformed into the final format and structure. This process is popularly referred as ELT (Extract, Load and Transform).

3. Storage

The data warehouse is a normally structured relational database that must be organized to hold information in a structure that best supports not only query and reporting, but also advanced analysis techniques, like data mining. Most data warehouses hold information for at least a year and some may also be used for

highly extended periods, depending on the business/operations data retention requirement. As a result a very large storage capacity may be required for the data warehouses. The retail and the telecommunications industries generally own very large data warehoused in the Terabytes (TB) range. The primary determinant of the size and shape of the data warehouse is the size and shape of the business problem. The size and shape of the data warehouse in a given enterprise is a function of the experience and maturity of the industry as to the use of business intelligence for decision support and competitive advantage as well as the length of storage.

4. Analysis Tools

The data in the data warehouse must be disseminated within the organizational employees for it to be useful. This information/knowledge dissemination can be performed by a very large number of software applications or can be tailor made to suit organizational needs. These applications include:

a. Business Intelligence Tools

These are software applications that simplify the process of development and production of business reports based on data warehouse data.

b. Executive Information Systems

These are software applications that are used to display complex business metrics and information in a graphical way to allow rapid understanding.

c. OLAP Tools

OLAP tools form data into logical multi-dimensional structures and allow users to select which dimensions to view data.

d. Analytical Applications

These are generally industry or domain specific applications that combine simple to complex ad-hoc reporting as well as simulation capabilities.

e. Data Mining

Data mining tools are software that allows users to perform detailed mathematical and statistical calculations on detailed data warehouse data to detect trends, identify patterns and analyze data.

5. Metadata

Metadata or "data about data" is used to provide pointers regarding that status and the information contained within a warehouse to its operators and users. It is also used as a means of integrating incoming data to the warehouse and further as a tool to update and refine the underlying warehouse model. Examples of data warehouse metadata include table and column names, their detailed descriptions, their connection to business meaningful names, the most recent data load date, the business meaning of a data item and the number of users that are logged in currently.

6. Operational Processes

Operational processes comprise of the tasks of loading, manipulating and extracting data from the data warehouse. It also covers user management, security, capacity management and related functions

7. Optional Components

In addition to the above mentioned primary data warehouse components the following components may be present in some data warehouses:

a. Data Marts

A data mart is a physical database (either on the same hardware as the data warehouse or on a separate hardware platform) that receives all its information from the data warehouse. The purpose of a Data Mart is to provide a sub-set of the data warehouse's data for a specific purpose or to a specific sub-group of the organization. A data mart is technically exactly like a data warehouse but it serves a different business purpose: it either holds information for only part of a

company (such as a division), or it holds a small selection of information for the entire company (to support extra analysis without slowing down the main system).

b. Logical Data Marts

A logical data mart is a filtered view of the main data warehouse but does not physically exist as a separate data copy. This approach to data marts delivers the same benefits as a physical data mart but has the additional advantage of not requiring additional disk space. Further it is always as current with data as the main data warehouse. However the primary disadvantage with this approach is that Logical Data Marts can have slower response times than physical ones.

c. Operational Data Store

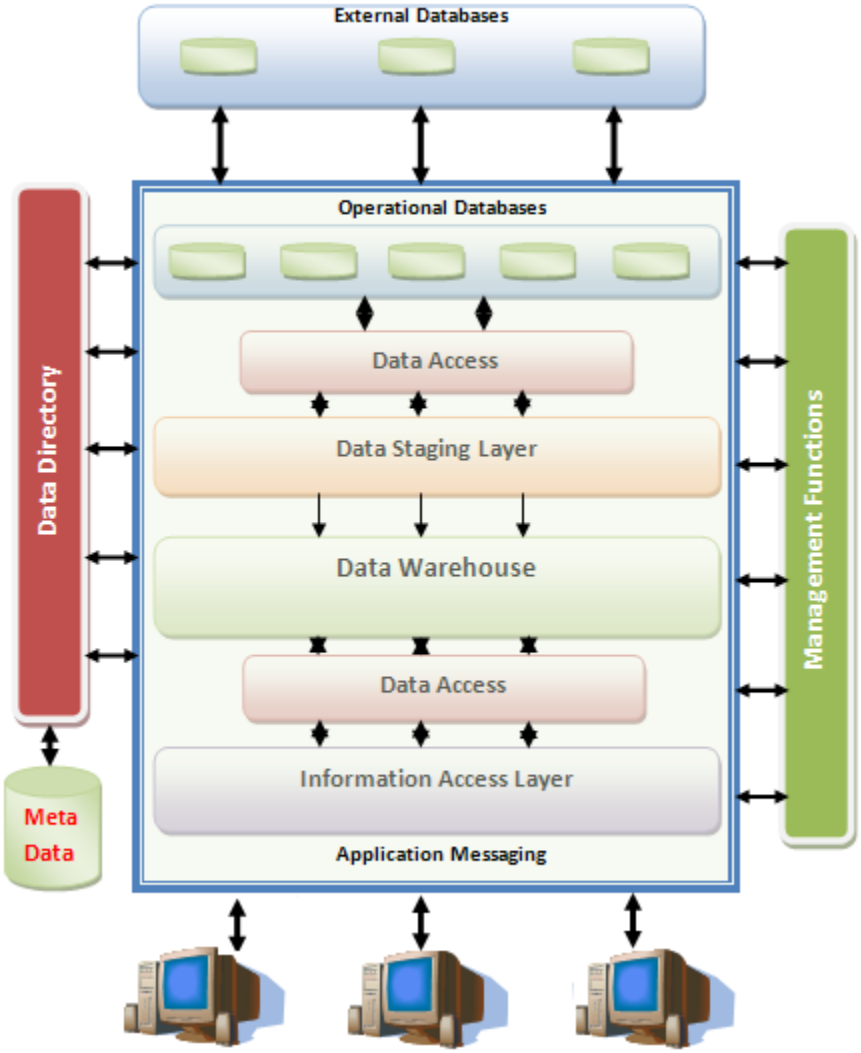
An Operational Data Store (ODS) is an integrated database of operational data. Its sources include legacy systems, and it contains current or near-term data. An ODS may contain 30 to 60 days of information, while a data warehouse typically contains years of data. They are employed in some data warehouse architectures to provide near-real-time reporting capability in the event that the Data Warehouse's loading time or architecture prevents it from being able to provide near-real-time reporting capability.

2.6 Data Warehouse Architecture

A Data Warehouse Architecture (DWA) is a method of representing the overall structure of data, including the communication framework, processing mechanism and presentation format that exists for end-user computing within an enterprise. The architecture is made up of a number of interconnected parts that includes the following:

Figure 2.5 Data Warehouse Architecture

Operational systems are employed to process data while supporting critical operational needs. The operational databases are historically created to provide an efficient processing structure for a relatively small number of well-defined business transactions. The limited



focus of operational systems makes it difficult for other management applications to interact with the operational databases. This difficulty in accessing the operational data is amplified by the fact that many operational systems are often 10 to 15 years old and the systems employed to access this data would

itself have become obsolete. The goal of data warehousing is to free the information that is locked up in the operational databases and combine it with information from other, often external, sources of data. Increasingly, large organizations are acquiring additional data from outside databases. This information may include demographic, econometric, competitive and purchasing trends.

The Information Access layer of the Data Warehouse Architecture is the interface with the end user. More specifically this layer represents the tools that the end-user normally uses day to day, e.g., Excel, Lotus Suite, Microsoft Access, SAS, etc. This layer also includes the hardware and software involved in displaying and printing reports, spreadsheets, graphs and charts for analysis and presentation. The last few years have witnessed tremendous growth in the information access layer with the end users having access to powerful computing devices. Further sophisticated tools facilitate enhanced analysis and presentation of data. There however exists the complex issue of making the raw data contained in operational systems available easily and seamlessly to end-user tools. A work around to this problem is to deploy a common data language throughout the enterprise.

The data access layer is primarily involved with facilitating the interface between the information access layer and the operational layer. The commonly employed data language is SQL, originally developed by IBM as a query language, which has become the de facto standard for data interchange. One of the key developments in the last few years has been the emergence of a series of data access filters like EDA/SQL that make it possible to access nearly all DBMS's and data file systems, relational or non-relational using SQL. These filters make it possible for the modern day information access tools to access the database management systems that contain historic data. The data access layer is vendor as well as protocol independent and spans different DBMS and file systems on the same hardware. One of the successful data warehousing strategy is to

provide end-users with "universal data access" wherein access any or all of the necessary data (subject to access privileges) is possible regardless of location or tools employed. This Layer is also responsible for interfacing access tools with operational databases.

In order to provide for universal data access, it is absolutely necessary to maintain some form of data directory or repository of meta-data information. Meta-data is the data about data within the enterprise. In order to have a fully functional warehouse, it is necessary to have a variety of meta-data available. This would include data about the end-user views of data and data about the operational databases. The end-users would be able to access data from the data warehouse or any of the organizational operational databases transparently without having to know where that data resides or the form in which it is stored.

The Process Management Layer is involved with scheduling the various tasks to be accomplished in building and maintaining the data warehouse and the associated data directory information. The Process Management Layer can be thought of as the scheduler or the high-level job controller for multiple processes/procedures required for keeping the contents of the data warehouse updated.

The Application Message Layer, functioning as a middleware in the warehousing system, is responsible for information exchange within the enterprise computing network. This layer contains the networking protocols and can also be used to isolate operational as well as informational applications from the data formats on either end. Application Messaging can, which is the underlying transport system, can also be used to collect transactions or messages and deliver them to a certain location at a certain time.

The data warehouse represents the logical or virtual view of the underlying data. In many cases the data warehouse would not be storing the data which may be physically on a different platform altogether. The core of the warehouse is where

the actual data used primarily for informational uses resides. In a Physical Data Warehouse, copies, in some cases multiple copies, of operational and or external data are actually stored in a form that is easy to access and is highly flexible. In earlier days the data warehouses were stored on main frames. However in the current scenario they may be located on client/server platforms or in some case separate networks referred to as storage area networks (SAN).

The final component of the Data Warehouse Architecture is Data Staging. Data Staging is also referred to as copy management or replication management. It includes all of the processes necessary to select, edit, summarize, combine and load a data warehouse with information or access data from operational and/or external databases. Data staging normally involves the scripting of complex access routines or programs. However the emergence of data warehousing tools has simplified this process. Data Staging may also involve data quality analysis programs and filters that identify patterns and data structures within existing operational data.

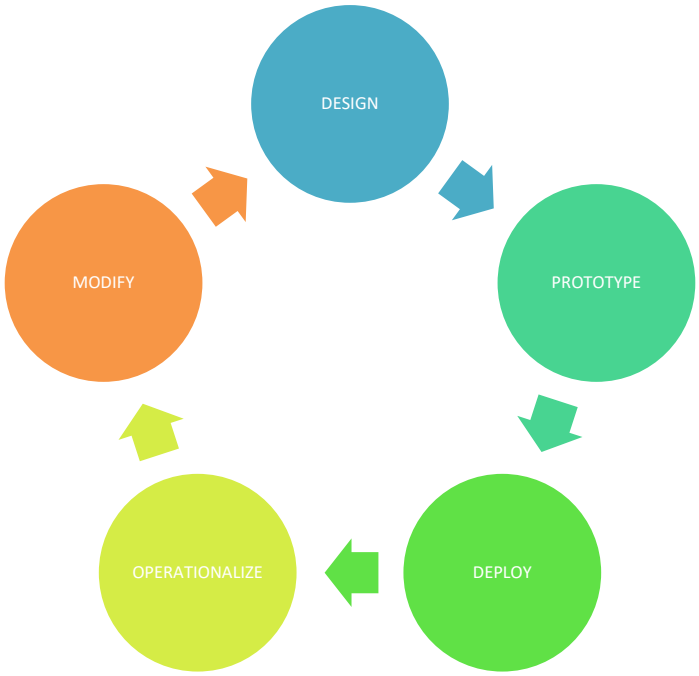
Every single data element in the physical design model is associated with a logical counterpart in the logical strategic information model. A critical part of designing the Data Warehouse architecture is reverse engineering the existing operational systems of record to match the physical design models in the architecture repository. This provides the basis for gap analysis and defining transformation requirements.

2.7 Data Warehouse Engineering – Life Cycle

The data warehousing lifecycle consists of five major phases, illustrated in the figure below that includes: Design, Prototype, Deploy, Operationalize and Modify. The life cycle commences with the design stage that includes the analysis of the existing enterprise data, evaluation of template data models, end user interviews to elicit operational requirements, identification of metrics related to the business, user and expected system performance and culminates with the design of the physical and logical schema. In the prototype development stage a

group comprising of members at all tiers of the organizational hierarchy including strategic decision makers, administrators and end users is formulated and their inputs crystallized to develop a working model of the warehouse or a data mart. This model is further used for gap analysis. After the approval from the target group the prototype is scaled up to an enterprise level or deployed at SBU level based on organizational requirements. This step is complemented by the development of extensive documentation, training, system integration, system optimization and allied activities. Stage 4 related to the routing maintenance of the warehouse along with the related activities of performance tuning and optimization as well as backup schedules. The final stage of the life cycle relates to the upgradation or changes to the warehouse architecture in response to changing business or technology landscape. This stage is continual in nature. The key activities in the engineering of a warehouse, and illustrated in the following figure 2.6 are as detailed below:

Figure 2.6 Data Warehousing Life Cycle



An important constituent of the engineering life cycle for a data warehouse is the enterprise need identification. The design of a data warehouse requires inputs from multiple individuals as well as user and management groups as opposed to routine operational systems with only single user group involvement. This can lead to the emergence of multiple and often divergent views or inputs related to the data warehouse design. This calls for the precise or accurate identification of the enterprise needs which is critical to the success of the data warehousing system.

An organizations strategic business plan would form the basis for the warehouse strategic information needs assessment. This information can be gathered through interviews with key enterprise managers and analysis of other pertinent documentation available. The commencement of the warehousing project should be followed by a series of facilitated focus group sessions to refine or modify the preliminary enterprise information that was elicited from business plans and executive interviews. These discussions would involve the corporate decision-makers who would be among the primary targeted end user groups.

The enterprise warehouse definition measures include the definition of time cycles or periods for the assessment, processing as well as the storage of enterprise warehouse data. Depending upon the operational domain of the organization under consideration this could be daily, weekly, fortnightly, monthly, quarterly, yearly or any other intermediate period. This time definition would vary between organizations. For example the Government of India (GOI) - Ministry of Economic Affairs in determining the economic trends would employ monthly, quarterly and annual timelines. In contrast the Ministry of Health and Family Affairs would require access to historical data, which may range from couple of years to decades, to formulate meaningful statistics for examination of the population explosion. A telephony service provider may employ hourly measures and may retain only few weeks of information on operational systems.

Once the enterprise needs identification is complete it is a good practice to ensure their communication throughout the enterprise. This ensures that the

enterprise needs including the process definitions and the critical success factors are made transparent throughout the organization. Thus all the organizational knowledge workers are precisely aware of the metrics governing the success of the warehousing projects along with its measurement parameters. This process would also be beneficial to the organizational knowledge workers who would not be directly involved in the warehousing project but would be involved in the backend processing or providing the raw data for the warehouse. These individuals would also be able to provide valuable feedbacks that could be used to refine the measures further.

There exists a possibility for organizational data conflicts that may exist due to ambiguity in the usage of key terminologies. Most of the organizations do not have a well defined nomenclature to be followed by all the departments and SBU's. To illustrate this point let us take the example of a service organization – The term 'customer' may imply an organization for the sales department, an organization or an individual who is being billed for the accounting department, an individual for the product development team. A well defined Data Warehouse model, cannot allow data definition conflicts that may arise due to the usage of homonyms, synonyms or any other naming terminologies, throughout the organization. Thus in the above example there cannot be an enterprise level definition for the term 'customer'. It is therefore imperative to provide a clear and unambiguous definition for every warehouse data entity along with a description of its usage, methods of derivation, categorization and timelines. These activities are critical to building a clear understanding of an enterprise's measures. The resulting enterprise architecture model, which links the enterprise needs with the warehouse data entities and enterprise rules, can be used as the basis for documentation and information dissemination to the to the enterprise users.

The data warehouse architecture design commences after the definition and the subsequent documentation of the enterprise needs measures and critical success factors. This activity necessitates the active involvement of the user groups through facilitated design sessions. The process initiates with the

definition of the warehouse metadata. In lay man terms metadata can be defined as ‘Information about the data’ in the warehouse. The important characteristics of the warehouse metadata are that they should be contextual (related to time), accurate, highly reliable, versioned (ability to archive data) and also include metadata about the quality of the warehouse data. The two primary types of Data Warehouse metadata are as outlined below:

i) Structural Metadata

Enterprise measures provide an insight to the data entities to be included within the warehouse. It also helps in identifying the data entities that need to be aggregated. The number and the types of data aggregation categories in a warehouse will depend directly upon the organizational requirements. This is explained with the help of an example in the paragraph below. Structural data is used to create and maintain the data warehouse.

Structural Metadata is used to represent the structural framework of a complex object. This representation could be physical and/or logical. A good example would be the organization of the contents of this book into chapters. The boundaries over the coverage of the chapters are fixed along with the relationship with the other chapters. The structural metadata completely describes the structure of the warehouse along with the associated content. The starting point for the building of structural metadata is the listing of data entities along with their characteristics and inter-relationships.

The type of individuals that provide inputs to the warehouse design also plays an important role in the final outcome of the warehouse structure and specifically on the number of aggregation categories. Individuals representing the top management represent “Strategic thinking” and are interested in higher level abstractions of data. These individuals require answers to broad level questions or the ‘abstract view and hence require only a very few aggregation categories. The procedures to ‘roll-up’ or abstract or aggregate data based on strategic inputs can however be complex. In sharp contrast individuals representing the

‘line function’ or the operational workforce of an organization require access to every measure grouped by category that employed in their domain. These individuals therefore require access to a fairly large numbers of aggregation categories. However the complexity of these categories would be less.

The system of record for the data entities within the data warehouse are identified by the structural metadata. In addition the logic for integration and transformation logic for moving an entity from its system of record to the data warehouse is also described. Further the schedule for refreshing or updating each of the data entities along with their archive requirements are also specified by the structural data. From this discussion it is fairly evident that a change in the data entity would result in a corresponding change in the structural metadata. However, as listed at the beginning of this section, it is important or even mandatory to retain the information regarding the changes and provide appropriate access to them as required. This is required to ensure that the structural metadata is contextual.

In addition to the information listed above, the structural metadata would also include performance metrics for applications and queries. These metrics are employed by developers for time estimation (length of time to execute a query or run a warehousing application) as well as performance optimization of the data warehouse. These metrics are also used for performance optimization of a Data Warehouse.

ii) Access Metadata

The dynamic linkage between a data warehouse and the end user applications, which interact with it, is provided by the access metadata. The enterprise nomenclature (standard terms, user defined terms and aliases) along with the enterprise measures are represented by the access metadata. The information regarding the location of the warehouse servers along with their description, databases, detailed data as well as the summaries are included within the access metadata. In addition the rule for navigating the enterprise dimension

views (drilling up/down) along with the subject hierarchies (e.g. product, channels or customers) are also provided by the access metadata while also supporting custom user defined queries and/or computations. The access metadata also includes access restrictions governing the display, modifications, analysis or distribution of standard as well as custom queries, computations or summaries.

The enterprise data warehouse architecture also defines or identifies the source of raw data that would be moved into the warehouse while also ensuring the consistency of the data entities and the transformation routines. This identification of the data sources (systems of record) of the warehouse data serves as a validation point for the enterprise measures.

A robust technology framework is a precursor to a stable organizational warehousing platform. The technology framework includes both the hardware as well as the software architecture. The hardware architecture includes the server environment, their placement, dedicated communication lines, backup mechanisms, storage devices and configurations (RAID, SAN…), authentication framework (centralized setup versus distributed, usage of authentication servers like RADIUS) as well as the backup power supplies. The software architecture would dictate the choice of the operating system environment, application software including the analytical software (Client/Server environment) and multi dimensional access technologies. .It would also include the applications or programs required to move the data into the warehouse, transformation routines and access control mechanisms along with the user interfaces and applications. Some of the key considerations for determining a suitable hardware platform the warehouse include the following:

- The size of the warehouse
- Platforms to be supported and scalability issues
- System optimization and performance
- Application support

The raw operational data cannot be directly loaded onto a warehouse. These application specific data must be converted to enterprise specific data before they can be used by end users of the warehouse. This is performed by the integration and transformation routines. The data from the sources identified have to be initially populated onto the data warehouse. Subsequently the contents of the warehouse need to be frequently updated. This process is achieved through the usage of integration and transformation programs that pull out data from the organizational operational as well as archival databases and systems. A single program can be employed for the initial population of the data warehouse as well as the periodic updation. However there may be certain cases that may warrant the use of separate programs. This is especially true in cases where the warehouse initial load, from operational systems, is very extensive and may severely degrade the performance of the existing users logged in. Also in case the warehouse updates are not too frequent and not significantly large it may be good option to use separate programs. Further the loading of warehouse data from archival systems as well as historical data from operational data systems is usually a non recurring activity and is done using separate programs/routines. The standard industrial practice is to use a set of programs for initially populating the warehouse and another set of programs to ensure its periodic updation. As is obvious, the programs for updating the warehouse are simpler than the load programs, and hence consume lesser system resources. In many cases the updation routines may be actually built onto operational systems to ensure the automatic updation of changes in a real time fashion. Many organizations opt for in-house development of the integration/transformation routines to ensure tighter integration with their operational systems. This option also helps organizations develop extensive documentation which may be helpful in case reconfigurations are required at a later date. In contrast the off-the-shelf programs may not be fully documented, be difficult to integrate with enterprise systems and difficult if not impossible to reconfigure.

In contrast to organizational databases the enterprise data warehouse is read-only. This implies that developers need not be concerned with the management of creation, updating and deletion capabilities. There however exists a need address the trade off between protecting the organizational intangible assets against unauthorized access while fostering effective organizational information dissemination to ensure effective utilization of knowledge resources. One of the solutions is to have an access controlled environment with different levels of users accorded a differing set of privileges based on need. For example access to the base data of the warehouse may be restricted to only administrators and/or developers while the other users would have access to derivations and summaries. In addition to access security, an enterprise must be concerned with physical security for its Data Warehouse. Because its contents are an extremely valuable organizational intangible resource, they must be protected against loss and damage. This protection is available in many forms ranging from simple backup and off-site storage strategies to installation of uninterrupted power supplies to the deployment redundant array of disks (RAID) for storage.

The data from the warehouse is accessed by users to generate useful information through well designed user interfaces. These user interfaces plays an important role altering the perception of the end users about the data warehouse. The guiding criteria for developing an effective user interface is its simplicity or are ease of use and performance. The deployment of graphical user interfaces (GUI) facilitates the development of a menus based hierarchical interfaces that is simple to use. In order to ensure enhanced performance developers must ensure that the hardware/software platform fully supports and is optimized for every chosen user interface. One of the prime selection criteria for user interfaces is the analysis of the information needs and the level of computer literacy of potential users. A good ‘rule of thumb’ is to employ simple and highly graphic interfaces for users who require access to highly summarized data while providing detailed data users a more complex but less graphical tools. The final requirements would be supports to the warehouse access metadata. An optimal user interface facilitates fast information retrieval in the desired format.

2.8 Recommended Reading

Books

1. Hoberman Steve, Data Modeling Made Simple: 2nd Edition Technics Publications, ISBN-13: 978-0977140060

2. Carlis Vincent John, Maguire Joseph and Carlis John, Mastering Data Modeling, Addison-Wesley Publishing Company, ISBN: 020170045X

3. Hirschheim Rudy, Klein K Heinz and Lyytien Kalle, Information Systems Development and Data Modeling – Conceptual and Philosophical Foundations, Cambridge University Press, 2008, ISBN-13: 978 – 0521063353

4. Reingruber Michael, Reingruber and Gary Gregory, The Data Modeling Handbook: A Best-Practice Approach to Building Quality Data Models Book Description, John Wiley & Sons, 1994, ISBN-13:978-0471052906

5. Simpson Graeme and Witt Graham, Data Modeling Essentials, 3rd Edition, Morgan Kaufmann, 2004, ISBN-13: 978-0126445510

URL

http://www.agiledata.org/essays/dataModeling101.html

http://www.1keydata.com/datawarehousing/toolreporting.html

http://www/inderscience.com

http://www.oracle.com/technology/books/pdfs/powell_dwtuning_ch01.pdf

http://en.wikipedia.org/wiki/Data_modeling

http://en.wikipedia.org/wiki/Data_analysis

http://www.agiledata.org/essays/agileDataModeling.html

http://www.amazon.com/exec/obidos/ASIN/0932633293/ambysoftinc

http://www.amazon.com/exec/obidos/ASIN/0387229507/ambysoftinc/

2.9 Referred Standards

Industry Standard Data Models (ISDM)

Method for an Integrated Knowledge Environment (MIKE 2.0)

2.10 Key Terms [Nomenclature]

Access Metadata

Application Messaging Layer

Business Area Models

CASE

Class

Data Access Layer

Data Directory Layer

Data Marts

Data Mining

Data Staging

Data Transformation Layer

Data Warehouse Architecture (DWA)

Data Warehouse Layer

Decision Support Systems (DSS)

EDA/SQL

Enterprise Data Model

Explicit Metadata

Exploratory Data Analysis (EDA)

Extract, Load and Transform (ELT)

Homonyms

Information Access Layer

Information Technology (IT)

International Business Machines (IBM)

Knowledge Discovery

Logical Data Marts

Logical schema

Metadata

Multidimensional Analysis

Online Analytical Processing (OLAP)

Online Transaction Processing (OLTP)

Operational Data Store (ODS)

Operational Database

Physical store

Pivoting

Process Management Layer

Query & Reporting

RADIUS

RAID

Storage Area Networks (SAN)

Small Business Unit (SBU)

Statistical Data Analysis

Subject Area Models

Summary Model

Synonyms

Terabytes (TB)

2.11 Summary

This chapter laid the foundation for preparing the blue print for designing a data warehouse. It provides an introduction to the commonly used data analysis techniques and their linkage with the data warehouse architecture. The key learning's from this chapter are as listed:

- A data warehouse is designed to provide easy access to high quality data sources. An important thing to note is that a warehouse is not the end objective, but rather the path leading to the end point. The end point is subsequent application of analytical and decision making tools to garner valuable insights from the extracted data.

- Multidimensional analysis is employed to extend the capabilities of query and reporting tools. This technique employs the use of structured data to enable fast and easy access to answers to typical questions as opposed to submitting multiple queries.

- An organizational data model does not change drastically over a length of time unless there is a fundamental change in the organization vision/mission. The data usage and the processes involved can however vary greatly across

enterprises within the same industrial domain, despite having commonality in their data requirements. This similarity of data facilitates the development of template data models that can be adopted by organizations operating in a similar domain.

- A Data Warehouse Model represents the actual organization of data within the warehouse. It describes the data structures and their inter-relationships. The domain specific organizational informational requirements are represented by the data model.

- A data warehouse forms the primary repository of an organization's historical data or acts as the corporate memory of the organization. A data warehouse is optimized for being integrated with transaction processing systems.

- The Data Warehouse Architecture represents the overall structure of data, including the communication framework, processing mechanism and presentation format that exists for end-user computing within an enterprise.

2.12 Check Your Learning

Review Questions

1. A data warehouse is designed to provide easy access to high quality data sources:

a. False

b. True

2. An organization requiring quick information retrieval (Query & Reporting capability) would deploy a model that structures the data in a:

a. Normalized Form

b. De-Normalized Form

3. A _____would be more appropriate if the objective is to perform multidimensional data analysis.

a. Enterprise Data Model

b. Subject Data Model

c. Dimensional Data Model

d. Logical Data Model

4. The most commonly employed data analysis technique is:

a. Query & Reporting

b. Data Mining

c. Multi Dimensional Analysis

d. KDD

5. Data mining is a data analysis technique, which is very different from query and reporting well as multidimensional analysis :

a. True

b. False

6. The different types of statistical data analysis techniques include:

a. Linear and nonlinear analysis

b. Regression analysis

c. Multivariate analysis

d. _____

7. The _____ are fully functional pre-designed data models built for a specific industry

a. Enterprise Data Models

b. Template Data Models

c. Subject Data Models

d. Physical Data Models

8. The _____ is primarily employed for strategic planning as well as disseminating the warehouse data requirements throughout the enterprise.

a. Enterprise Data Model

b. Template Data Model

c. Subject Data Model

d. Physical Data Model

9. The _____are detailed data models representing standard functions within a specific business domain.

a. Enterprise Data Models

b. Template Data Models

c. Subject Data Models

d. Business Data Models

10. The _____ describe functional subject areas that are unique to an organization and represent the lowest levels of data and provide the design foundation for the data warehouse, data marts, applications development, business analysis and strategic planning..

a. Enterprise Data Models

b. Template Data Models

c. Subject Data Models

d. Business Data Models

_____
__

Exercises

1. Write a brief note on the commonly employed data analysis techniques and their significance.

2. What are Data models? List down three data models with a brief explanation for each.

3. List down the key steps in data modeling along with a brief explanation.

4. Write short notes on the following:

- Data Marts
- Data Warehousing Life Cycle

5. Enumerate on the structure and composition of a data warehouse.

_____

__

Research Activities

In continuation to the research activity listed in Chapter 1, you are requested to:

Choose an appropriate data analysis technique and data model with due justification provided for their selection

Prepare a data warehousing life cycle detailing the major phases

_____

__

Chapter Objectives

The objectives of this chapter are:

To discuss the architecture and implementation choices available for designing and developing data warehouses and data marts

Chapter 3

Data Warehouse Architecture & Design Considerations

3.1 Introduction

This chapter presents the issues that govern the choice of data warehouse or data mart architecture. Data marts refer to smaller data warehouses that can function independently or can be interconnected to form a global integrated Data Warehouse. The choice of the warehouse architecture is done prior to beginning implementation. The architecture can however be modified, after the implementation commences. The choice of the warehouse architecture is generally a management decision that is based on multiple factors. These include the organizational IT infrastructure, business landscape, preferred warehouse management and control structure, high level dedication to the implementation effort, and implementation scope, capability and maturity of the technical environment and financial commitment to the project. The implementation approach selected can have a dramatic impact on the success of a data-warehousing project. The variables affected by that choice include completion time, return-on-investment (ROI), time to realize organizational benefits of warehouse implementation, user satisfaction, potential implementation rework, future resource requirements and the data warehouse architecture selected.

3.2 Data Warehouse Architecture [1]

The selection of the data warehouse architecture along with the physical locations of the warehouse and data marts (as the case may be) and the control/access mechanisms are interlinked. For example, the data warehouse can be centrally located and managed or distributed geographically. The control can however be centrally or localized. This chapter presents three broad data warehouse architectures along with the implementation guidelines. The implementation choices include the top-down approach, bottom-up approach, or a hybrid approach. In fact the warehouse architecture could also be a combination (hybrid) of the three broad types presented in this chapter. For example, an organization can employ physically distributed warehouse architecture with centralized access and control. The warehousing implementation could however be from individual departments or SBU with their own data marts servicing their unique requirements.

3.2.1 Global Data Warehouse Architecture [2]

A global data warehouse is designed at an enterprise level with full integration across the various organizational departments and/or SBUs. Each department/SBU has complete access to the contents of the warehouse. A global warehouse is designed and deployed to meet the organizational information requirement and functions as a common repository for the organizational decision support data.

It is generally perceived that a global data warehouse architecture, location and control are centralized. However the ‘Global’ prefix is used to indicate organization wide usage and access and is not indicative of the physical architecture. Such warehouses can be physically centralized or distributed throughout the

organizational geographies. A physically centralized global warehouse resides in a single location while a distributed global warehouse is also to be used by the entire organization, but it distributes the data across multiple physical locations within the organization. The management of local warehouses may be split across multiple IT teams at the local geographies. The management may be done locally while the control is centralized. For example, the physical locations of the distributed warehouses or data marts could be based on organizational departments and SBUs with local access policies. The local departments/SBUs decide on the contents of the warehouse, the frequency of data updation and user access (access by other departments and SBUs) to the warehouse. The server administration and communication infrastructure would be managed by the IT department.

Data for the data warehouse is generally gleaned from operational systems and sources external to the organization (depending upon requirements). It is then filtered to eliminate any irrelevant data and transformed to meet the data quality and standard requirements specific to the organization. The data is subsequently loaded onto the data warehouse for access by end users.

The Global warehouse architecture enables end users to have an enterprise wide or organizational view of the data. However an organization should be very clear about the need for an enterprise data since the implementation and operational costs involved are very high.

3.2.2 Stand-Alone Data Mart Architecture [3]

This type of architecture refers to stand-alone or independent data marts that are controlled by a particular workgroup, department, or SBU and are built solely to

meet their specific requirements. In many cases such data marts would not require any connectivity with similar data marts strewn across the various departments or SBUs within an organization. The data residing in these marts would be generated internally or may be extracted from operational systems and also from sources external to the organization. The stand-alone data mart architecture requires additional technical skills to implement, but the resources and personnel could be owned and managed by the workgroup, department, or line of business. These types of implementation typically have minimal impact on IT resources and can result in a very fast implementation. However, the minimal integration and lack of a more global view of the data can be a constraint. That is, the data in any particular data mart will be accessible only to those in the workgroup, department, or line of business that owns the data mart.

3.2.3 Interconnected Data Mart Architecture

A distributed implementation of inter-connected data marts is referred to as interconnected data mart architecture. An independent data mart is implemented for each organizational work group, department or SBUs. These independent data marts are interconnected and integrated to provide an enterprise wide or corporate wide view of the data. A fully integrated data mart is equivalent to a global data warehouse. Users in one department would have access to their departmental data mart along with the capability to access and work with data marts in other departments. This architecture brings with it many other functions and capabilities that can be customized on a user or department basis. However these choices can bring with them additional integration requirements and complexities as compared to the independent data mart architecture. For example, the access, administration and manageability of the warehouse need careful planning and consideration in addition to employing a common data sharing schema across the organizational data marts. Alternatively another tier in the architecture would have

to be included to host data common to multiple departmental data marts. These requirements add a degree of complexity to the architecture. This is however offset by the significant benefits that can be accrued owing to the more global view of the data. Interconnected data marts can be independently controlled by a workgroup, department, or SBU. This includes the selection of data sources, the frequency of updation, user access, tools and the physical placement of the data marts. The role of the IT department is thus restricted to providing network connectivity, backup & restore capabilities and the server security administration.

3.3 Implementation Approaches

The issues that affect the choice of an approach can be primarily narrowed down to deciding between addressing short-term tactical departmental requirements and long-term strategic organizational needs and the choice of a suitable data architecture.

A Top-Down approach can yield the best long-term results at the organizational level. However this approach is complex, involves higher implementation costs and its success is dependent on evolving a single, consistent, accepted and valid view of the business and the underlying data. This would involve (depending upon the organizational size and structure) a lot of maneuvering to accommodate specific departmental or SBU requirements while evolving an organizational data standardization program.

The Bottom-Up approach favors the use of smaller, more focused applications of Warehouses that can avoid the disadvantages of the Top-Down approach by simply limiting the extent of the implementation. This approach also exhibits simpler data archaeology problems: there are usually limited data sets, limited user views, a good understanding of the data needs and how they relate to the business problem. This approach presents a tradeoff between the disadvantages of dealing with data standardization issues for the longer-term inability to operate at an

enterprise level. In this approach, each department is responsible for extracting whatever data they need, defining their own metadata and using their own private Warehouses for decision support at the departmental level. The common problems encountered with this type of approach include lack of enterprise wide scalability, lack of standardization and the issue of different answers being generated for the same questions. The primary reason of this lies in the underlying philosophy of the Warehouse architecture. Each mart can create confusing, overlapping and contradicting views of the business. This approach works if the organization has a business problem with a single focus and the data to solve that problem exists in only a few places, with no political ownership issues.

Some organizations use a hybrid approach to gain the speed and cost advantages of the highly focused departmental approach, yet at the same time making sure that the implementation is consistent with the overall goals of the organization. This approach uses the principles developed in Rapid Application Development (RAD) methodologies and intentionally delivers iterations of the departmental Warehouse, attempting at each iteration to come closer to an overall enterprise data model and data architecture.

This allows the organization to take advantage of the speed and cost savings of the smaller approach while at the same time mitigating and eventually overcoming the lack of integration and islands of automation problems inherent in this approach. Other organizations are using the smaller implementation as a proof of concept and prototype/pilot installation. This assists in proving the benefits of the technology on a smaller scale, and smaller risk, and then scaling the solution into a more global implementation, either by building more integrated departmental Warehouses or by moving to a full global enterprise Data Warehouse implementation. Any of these approaches can work for an organization. It is important to be aware of the issues associated of each approach and actively take proactive steps for its mitigation.

3.4 Implementation Strategies

The implementation choices discussed in this section offer flexibility in determining the criteria that is important in any particular implementation. The choice of an implementation approach is influenced by such factors as the current IT infrastructure, resources available, the architecture selected, implementation scope, the need for more global data access across the organization, ROI and finally the speed of implementation.

3.4.1 Top-Down Implementation [4]

A top down implementation requires more planning and design work to be completed at the beginning of the project. This brings with it the need to involve people from each of the workgroups, departments, or SBUs that will be participating in the data warehouse implementation. Decisions concerning data sources to be used, security, data structure, data quality, data standards, and an overall data model will typically need to be taken and documented prior to the commencement of the actual implementation process.

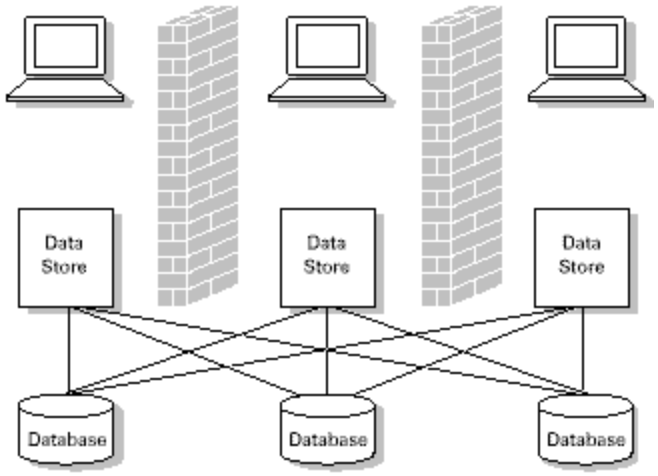


Fig 3.1 Top-Down Data Warehouse Architecture [1]

The top down implementation can also imply more of a need for an enterprise wide or corporate wide data warehouse with a higher degree of cross workgroup, department, or line of business access to the data. This approach is depicted in the figure 3.1 above. As illustrated in the figure a top-down approach is typically employed to structure a global data warehouse. If data marts are included in the configuration, they are typically built afterwards and are populated from the global data warehouse rather than directly from the operational or external data sources. The top-down approach necessitates the initial creation of a corporate infrastructure. A top-down implementation can result in more consistent data definitions and the enforcement of business rules across the organization, from the beginning. However, the cost of the initial planning and design can be significant. It is a time-consuming process and can delay actual implementation, benefits, and ROI. For example, it is difficult and time consuming to determine, and get agreement on, the data definitions and business rules among all the different workgroups, departments, and participating SBUs. Developing a global data model is also a lengthy task. In many organizations, management is becoming less and less willing to accept these delays. The top down implementation approach can work well when there is a good centralized IT organization that is responsible for

all hardware and other computer resources. In many organizations, the workgroups, departments, or SBU may not have the resources to implement their own data marts. Top down implementation will also be difficult to implement in organizations where the workgroup, department, or line of business has its own IT resources. They are typically unwilling to wait for a more global infrastructure to be put in place.

3.4.2 Bottom-Up Implementation [4]

In order to secure management as well as end user buyin most organizations desire to have a warehousing solution put in place within the least amount of time while demonstrating creditable ROI. Such a requirement can be satisfied by the design, development & deployment of small warehouses or data marts satisfying the requirements of a department or SBU. This approach is referred to as a bottom-up implementation and is used to kick start the warehouse development activity within an organization without the need for a global infrastructure. This global infrastructure can now be developed incrementally as initial data mart implementations expand. This approach is more widely accepted today than the top- down approach because immediate results from the data marts can be realized and used as justification for expanding to a more global implementation. The figure 3.2 below depicts this approach. In contrast to the top-down approach, data marts can be built before, or in parallel with, a global data warehouse. And as the figure shows, data marts can be populated either from a global data warehouse or directly from the operational or external data sources.

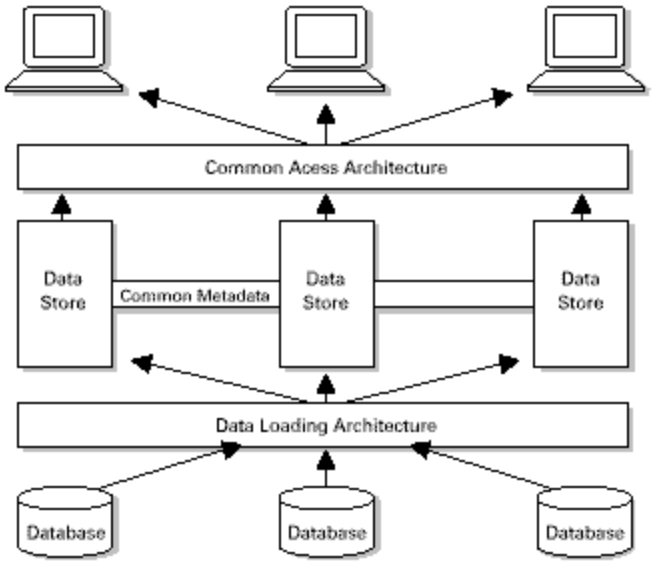


Figure 3.2 Bottom-Up Architecture [1]

A Bottom-Up Implementation commences with the creation of a single departmental/SBU data mart. Over a period of time the activity intensifies and results in the creation of data marts in other departments/SBUs. The bottom-up implementation approach has become the choice of many organizations, especially managed service organizations, because of the faster payback. It enables faster results since the data marts have a less complex design than a global data warehouse. In addition, the initial implementation is usually less expensive in terms of hardware and other resources than deploying the global data warehouse. Along with the positive aspects of the bottom-up approach are some considerations. For example, as more data marts are created, data redundancy and inconsistency between the data marts can occur. With careful planning, monitoring, and design guidelines, this can be minimized. Multiple data marts may bring with them an increased load on operational systems because more data extract operations are required. Integration of the data marts into a more global environment, if that is the desire, can be difficult unless some degree of planning has been done. Some rework may also be required as the implementation grows and new issues are uncovered that force a change to the existing areas of the

implementation. These are all considerations to be carefully understood before selecting the bottom up approach.

3.4.3 Hybrid Approach

As discussed in the preceding section there are advantages as well as disadvantages with the top-down or the bottom-up data warehousing implementation approaches. However in practice the best implementation approach, especially in the modern service organizations, is a hybrid approach combining the advantages of the traditional approaches. This can be a difficult balancing act but provides an optimal solution for very large enterprises. The approach requires planning at the global (enterprise) level along with the integration of the stand-alone data marts built at the departmental level (bottom-up approach). The key to this approach is to determine the design required to support integration as the data marts are being built with the bottom up approach. A base level infrastructure definition for the global data warehouse needs to be finalized at the business level. For example, the first implementation step would be to identify a department/SBU from which the development activity can be initialized. A high level view of the business processes and data areas of interest to the target department/SBU will provide the elements for planning the implementation of the data marts. As data marts are implemented, a plan to handle the data elements required by multiple data marts can be developed. This could herald the movement to a global data warehouse structure or simply a common data store accessible by all the data marts. It some cases it may be appropriate to duplicate the data across multiple data marts. This is a trade-off decision between storage space, ease of access, and the impact of data redundancy along with the requirement to keep the data in the multiple data marts at the same level of consistency. There are many issues to be resolved in any data warehousing implementation. A combined approach can enable resolution of these issues as

they are encountered within the narrow confines of a data mart rather than a global data warehouse. Further careful monitoring of the implementation processes and management of the issues could result in gaining the advantages of both the implementation techniques.

3.5 Infrastructure Requirements & Design Issues

The overall data warehouse architecture consists of data sources, decision support systems (DSS), application components along with the hosting, supporting and connectivity infrastructure, as highlighted in the figure 3.3 below:

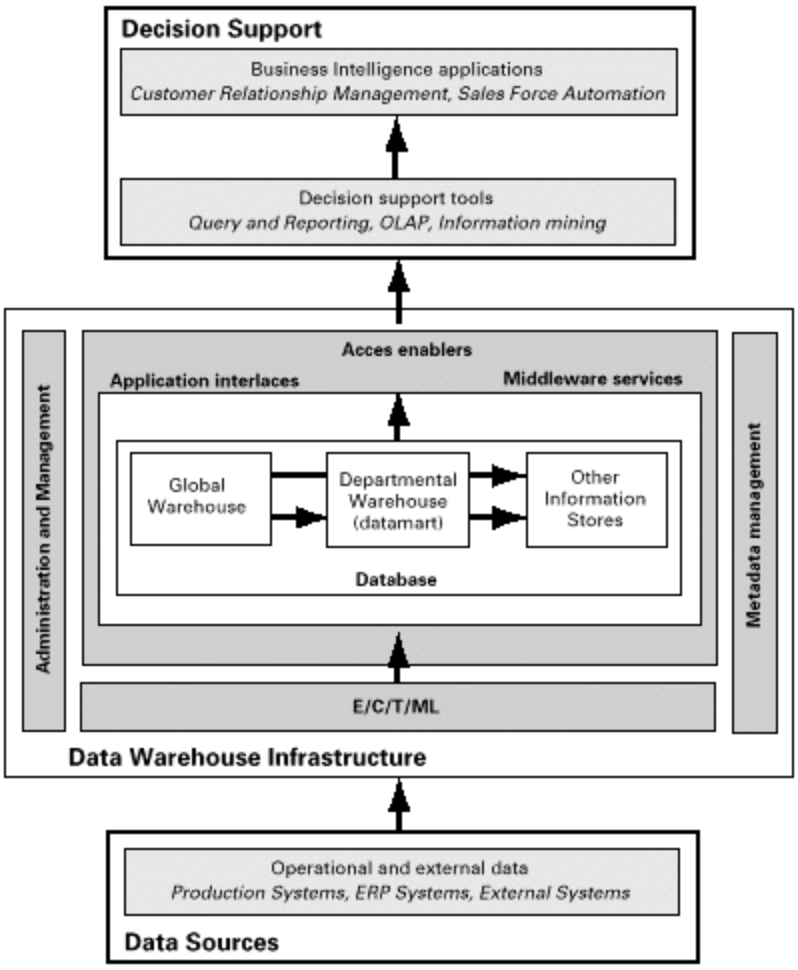


Fig. 3.3 Data Warehouse Infrastructure [1]

The data sources could be a multitude of information systems (IS) that provide raw data for the business intelligence (BI) applications and can include flat files, spreadsheets, IMS or IDMS and relational databases. This data can be found in production databases, online transaction processing systems (OLTP), enterprise resource planning systems (ERP) or external systems with proprietary storage mechanisms.

The Data Warehouse Infrastructure is made up of the components and methods necessary to massage the raw data from the source systems into refined data that can be used for building and deploying BI applications. ECTML (Extracting, Cleansing, Transforming, Moving and Loading) functions take raw data from the source systems and move it into the target enterprise data warehouse for management, administration, and potentially for staged distribution into dependent data marts. In the Decision Support part of the architecture, data can be accessed in the enterprise data warehouse using traditional ad hoc query tools or multidimensional query tools, or from a data mart focused on a specific BI application, such as Customer Relationship Management (CRM).

The ECTML portion of the Data Warehousing Infrastructure architecture is made up of Data Acquisition, Data Re-engineering, Data Loading, Meta-data Integration, and Warehouse Management functions. The data warehousing infrastructure portion of the architecture provides the connections that reliably and predictably ties together the vast amount of legacy data sources with the decision support systems (DSS) and BI application targets. This infrastructure is required when an organization initially builds and deploys a data warehouse. The infrastructure components are also required on an on-going basis to support the periodic refreshes of warehouse data, as well as to support business requirements and technological changes in the least disruptive manner.

The changing business landscape has resulted in the emergence of agile organizations that are constantly adapting to the changes in the external environment. Therefore the integration of new technologies onto the data-warehousing infrastructure to facilitate changes in accordance with the business landscape can be extremely challenging. The key is to develop a workable set of technologies, processes, and standards that will be the cornerstone for both the initial building and deployment of the warehouse and the subsequent development

of a set of BI applications. The following is the list of general assumptions for developing BI applications:

1. Source systems will change, because of new business requirements or because of acquisitions and mergers.
2. Target data warehouse and data mart platforms will change.
3. Any industry standards will be incomplete, and those that are available will be in a continual state of evolution.
4. Business user requirements will change and expand.
5. Data volumes will grow rapidly

The complexity and requirements of the ECTML component helps to highlight and evaluate the optimal approach to developing an enterprise warehouse along with its supporting components and applications. The following section presents the key issues in designing a warehouse.

3.5.1 Data Acquisition

Data acquisition involves finding the data that is needed for the enterprise warehouse from the vast number of potential data sources within the organization. A major problem faced by most organizations is the issue of data within legacy systems. In a large corporation there might be several systems with incompatible architectures, data representation schemes, and operational models. Many legacy systems are developed around flat files or network databases. For example the legacy systems employ EBCDIC codes while the modern day computing systems employ ASCII codes. In addition some of the earlier software packages were designed for endless flexibility, with near limitless possibilities for extensions supported by developing proprietary data management structures. This is in

contrast with the newer generations of systems implemented SQL-based relational structures.

3.5.2 Data Replication & Reporting

An organization may already have some extraction jobs that replicate part of the source data for sharing among several applications. For example, a customer service help desk may take regular downloads from a master customer file. These issues must be factored into the warehouse design. In addition management reports which are generated on a daily, weekly, monthly, quarterly or annual basis which occur on a regular or ad hoc basis contain a lot of information and knowledge over a period of time. This information must be captured and populated onto the warehouse.

3.5.3 Data Extraction Utilities [5]

Most database packages have import and export capabilities for exchanging data with other databases or applications. Data unload utilities are commonly used for extraction from a database. These utilities provide simple means for routine data extraction from the database. These pre-existing extraction utilities however may only be adequate for pilot implementations and may not be able to handle complex tasks like meta-data capture, transformation capability, and data cleansing. Enterprise data warehouse solutions necessitate the use of products specifically designed for data extraction and subsequent transformation. These off-the-shelf products facilitate the automation of the data retrieval process and the subsequent transformation and transport of data onto the warehouse. These programs also provides the functionalities for error logging and maintaining t also provides the ability to log the errors during the process as while maintaining the list of past efforts.

3.5.4 Data Quality, Cleansing, and Re-engineering

Once an extraction methodology is in place the next challenge is to identify a suitable method for cleansing and re-engineering the data. The population of enterprise data warehouses or data marts with usable, high-quality data can be highly complex. In order to avoid the problems arising out of the movement of irrelevant data onto the data mart or warehouse a set of a number of issues involving data acquisition and usage needs to be handled. These issues are highlighted in the subsequent sections.

Data Quality

One of the most common problems with effective data management is the lack of data consistency. The fragmented data, normally associated with legacy systems, introduces high levels of inconsistency. For example, legacy systems capture and process similar information, such as gender, marital status, date, etc. However, there may numerous mechanisms for structuring the data syntactically and semantically. Data may be captured using a number of different units of measure. Depending upon the type of system gender may be encodes as vide single letter designators (M/F) or binary designators (0/1) or full words (Male/Female) or the use of a ☑ against the appropriate option. In addition marital status, conventions for capturing titles (Jr., Mr., Ms.) can be sources of inconsistencies within the warehouse.

Another dimension of the data consistency problem is the issue of data reuse on legacy systems. Legacy systems normally do not facilitate data reuse. Customer or supplier names may have been entered differently across multiple systems preventing a consolidated view of the customer base or a near real time vision of

supplier business. This is especially pronounced if an organization has multiple business subsidiaries. Mergers & Acquisitions (M&A) introduces another set of inconsistencies related to various coding conventions. Customers, locations, products, and other business entities can be represented in an infinite number of ways. A single view of each entity or the ability to have a unique representation of all "instances" with a single set of attributes is a pre-requisite to effective information management. In the absence of a single view the meta-data would not be representative of the warehouse data. A thorough examination of the meta-data can divulge inconsistencies with the meta-field descriptions and business applications. Some of the common inconsistencies are as listed below:

1. Commercial names improperly mixed with personal names
2. Foreign names mixed with domestic names
3. Relationships that trickle over into address fields
4. Name fields with undefined relationships and location information
5. Address Fields
6. Extraneous noise (E.g. - An 8-character string within a 10-digit PAN No. field),
7. Addresses with missing pin codes
8. Truncated Information
9. Inconsistent use of white space, special characters, and field boundaries
10. Overlapping name and street address fields
11. Missing values in name and address fields
12. Abbreviations

Data Cleansing

Once consistent semantics and syntax have been established there is the potentially significant issue of data content. The status of the underlying data can

be provided through an automated exploration process. The consistency of the data is established by the use of various techniques including data cleansing, parsing, lexical analysis and probabilistic matching. Pre-built procedures can be used to parse and analyze rules and general fields (without any restrictions) to identify related records without any common key exists, while constructing new records based upon new sets of data resolution and consolidation rules. These procedures automate the analysis, comparison, resolution (reconciliation) and standardization of data records. Subsequent to the completion of the data cleansing activities the most effective method for moving and loading the data onto the target data-warehousing platform must be identified.

Complex hierarchical structures within a organization can provide a challenge to organizations needing to customize data for specific departments, such as creating a data mart for the Marketing department. Ideally these data marts would be sourced from the enterprise data warehouse where the information has been cleansed and an enterprise data model for the corporation has been established. However, many businesses opt for independent data marts as an expeditious path to a short-term ROI. This issue merits careful consideration in order to ensure that each data mart does not require its own independent extract and load from each of the legacy databases. It is expensive to replicate the extraction and cleansing process for each additional data mart that gets placed into production. Also the cycles required on the legacy systems for servicing the multiple overlapping requests for data extractions will impact the availability of the legacy system.

Data Re-engineering

Enterprise data warehouse systems need to be available to support organizations BI applications. This means that the performance of the warehouse load process

can become a critical factor in achieving the availability requirements. When loading large amounts of data, a special focus on I/O performance and data movement is required to ensure minimal load, build, and indexing time. Native database bulk load facilities generally provide a mechanism that requires the data be formatted prior to loading. This activity might add to the overall build and deployment effort. Generally, bulk data loaders or native programs with embedded database provide the best performance. The performance characteristics of the target database and warehouse platform need to be carefully evaluated. The build/load/index tradeoffs to develop an optimal model for moving the refined data into the data warehouse needs to be identified and balanced.

There are a number of vendors that provide data acquisition and loading solutions, either as stand-alone specialized products or as part of an overall suite or solution. There are single-vendor solutions and multi-vendor interoperable solutions. A single vendor solution means that the vendor provides all of the components for the solution and is responsible for the component-to-component communications. A multi-vendor interoperable solution involves a set of products from multiple vendors which communicate via a set of standard interfaces, provide their own specific set of complementary features and functions, and share appropriate technical meta-data.

The success of the single-vendor solution depends upon the ability to control the source systems associated with the data warehouse. Another is the commitment of the vendor to maintain and enhance their components to remain current with the state-of-the-market. Unfortunately, in a complex environment these single-vendor solutions tend to break down in both areas IT has little or no control over the source systems and few vendors have the technical resources to stay at the head of the market in all of the component technologies.

In case of multi-vendor solutions, the products that meet the organization's requirements for system availability, reliability, scalability, and quality are selected for their ability to easily respond to changing business requirements. The success of this solution depends on specific vendor support for inter-component communications standards. In the absence of support for standards proprietary vendor interfaces would have to be deployed. This may cause integration support issues in the long run.

3.5.5 Warehouse Scheduling and Management

Warehousing scheduling and management is accomplished by using products and procedures to control the Data Acquisition, Data Re-engineering, and Data Loading processes. The objective is to develop an operational environment for data warehousing that meets the organization's requirements for system availability and data quality. In and operational environment, scheduling and management necessitates the combination of routine procedures with systems-level management products. A well designed scheduling function is transparent to the warehouse users while a poorly planned function results in the failure of the BI applications. The data warehouse scheduling process controls the overall data extract and load routines for enterprise data warehouses and dependent data marts, as well as the other system functions needed to provide a secure, reliable infrastructure. The primary issues tend to center on system dependencies, job dependencies, exception handling, and cross-platform control.

Scheduling is required to perform the initial data warehouse build and subsequently on a regular basis to ensure updated data. The frequency of data updation is required to be estimated on the basis of the BI applications installed. The primary considerations include the identification of the sources of fresh data and the frequency of extraction/updation. There are a lot of off-the-shelf products

that can be used directly to assist with the process of building and managing the processes associated with data warehousing. System dependencies occur when a downstream system, such as an extract routine, needs to wait for an upstream system, like a bank accounting system, to complete processing before triggering a checkpoints that flags the data to be in a consistent state and enables it to be moved onto the next process. Scheduling products typically accommodate system dependencies as nodes in a workflow, and can either plan for data to be available at a set time, or when the upstream system indicates that it is ready to deliver data Interfaces are periodic with periods ranging from hourly to daily, weekly, monthly or yearly. Even the best-designed and tested scheduled process is going to break down at some point. The scheduling product needs to handle these exception conditions either by being able to re-start the offending process, or at a minimum prevents data created by broken processes from being sent to downstream processes. The best products are able to re-start broken processes from the point at which they broke. This can be a tremendous time saver when a process terminates when it is 90% complete.

The scheduling product needs to be able to control the processes across the systems within its domain. A few organizations have Online Transaction Processing (OLTP), data warehousing & database systems integrated onto a single system platform. In such cases inter-systems communications facilities are needed to trigger and track processes that are executing on platforms across the environment. The latest products available currently in the marker incorporate object oriented technologies to provide applications capable of handling these challenges. Performance management (PM) & tuning, database administration, and security administration for OLTP systems and data warehouses. However backup and recovery functions needs to be handled differently. As in OLTP systems the data warehouse must be able to rollback and restart from some known point in time when the data was in a "current" state, with a known level of referential integrity and commit completion. Then a restore operation updates the changes to

the current "live" data. The data warehouse should have a backup and recovery strategy that will enable the organization to recover data in case of emergency. However the approach may be significantly different. In cases of a small departmental data mart re-running the extract and load routines off the production database may be faster and equally effective compared to a regular backup.

Disaster recovery mechanisms need to be carefully planned in a warehousing environment. Organizations are becoming more dependent on the information that a data warehouse provides to run the business. This places the importance of the warehouse application on the same level as other mission-critical business systems. The data warehousing requirements are highly different from those of OLTP applications, which are standardized, and require specialized considerations.

3.5.6 Meta-data Management

Meta-data is extremely crucial and has multiple uses in a data warehouse environment. Meta-data, or information about data are used by administrators and end users to provide them with data descriptions or informational objects that can be accessed in the warehouse. For example, profit might be the product of a complex series of calculations based on regional revenue and expense. Meta-data is used to document these calculations so they are well understood by the business user trying to gain a competitive advantage by mining information out the operational data. Meta-data also provides detailed descriptions of objects and actions that are used by the software tools that are deployed. Meta-data is typically stored in its own meta-data repository which is a key infrastructure layer that spans Data Sources, Data Acquisition, Data Re-engineering, and Data Loading, and Business Intelligence processes. The figure below provides an illustration of the use of meta data in an warehousing environment.

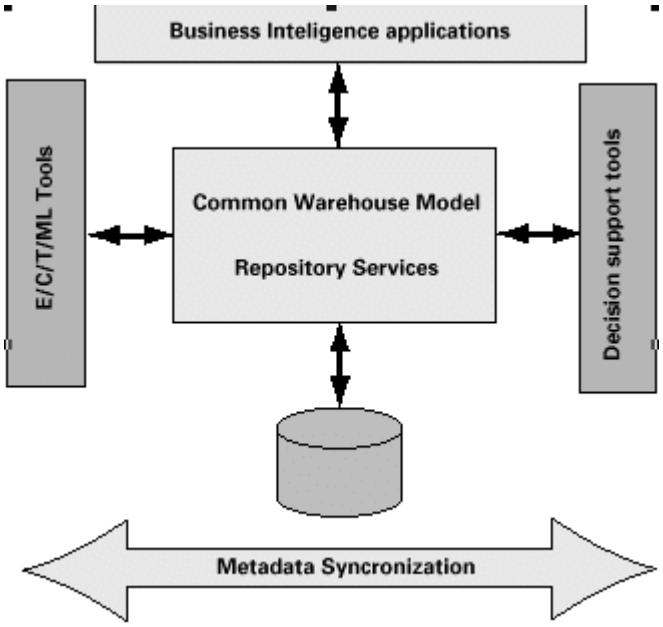


Fig 3.4 Meta Data [1]

Meta-data repositories are generally built as relational databases. However object oriented techniques are now employed to wrap meta-data definitions as objects. This helps solve many of the problems that are encountered in the relational-based models and allows data warehousing products to exploit traditional repository services such as versioning, security/access control, and change management. There are two types of meta-data:

1. Technical meta-data
2. Business meta-data.

Both types of meta-data are important in building, maintaining, and using a data mart or data warehouse. For business users, meta-data can provide information about the data in the data warehouse (such as what it means, how to access it,

and when it was last updated) as well as other related information (such as reports, spreadsheets and queries related to the data). For data warehouse administrators, meta-data is involved in all aspects of their job; meta-data defines, documents, and drives the processes of the data warehouse.

Meta-data helps to achieve two major objectives. It provides a means to improve the productivity of data administrators and the reliability of BI solutions. Many components are used when building the infrastructure for supporting BI applications. They include database management systems, modeling tools, transformation tools, process managers, and data mining and decision support tools. These tools leverage many different platforms, data formats, and vendor suppliers. Each tool uses its own meta-data store and administrative interface and using multiple as part of a data warehouse process necessitates the passing data and meta-data between them. Warehouse administrators must create and maintain such data bridges wherein similar information can be channeled onto multiple tools without affecting the consistency of the warehouse meta-data.

The second major objective of the meta-data is to provide a means to assist business analysts and end users in locating and understanding data. Users need to understand the lineage or genealogy of given piece of information along with its relevance.

Defining and maintaining meta-data is a major undertaking that will require broad enterprise support. As business requirements change, users are constantly changing their data and the way they need to look at the data. This puts an additional burden on the data warehouse to track and be aware of all the activity surrounding the operational/production systems, and to keep track of these changes, as they impact the meta-data. Keeping track of physical changes in the

data is relatively easy. However, if the changes involve semantics of the data, those changes are more difficult to identify and track. For this reason, it is very important that a data warehouse cross-functional team participates in tracking these types of system changes to maintain consistency in the meta-data definitions. This is particularly true in cases of enterprise reference data, where the semantics must be understood and accepted universally.

3.5.7 Industry Standards

Along with the ability to track meta-data changes the ability to share the changes with the platforms that access the repository are also required. Cross-industry standards groups such as the Meta Data Coalition (MDC) and the Object Management Group (OMG) have defined standards for exchanging meta-data. Similarly the Common Warehouse Metadata Interchange (CWMI) work group sponsored by the Object OMG is focused on establishing an industry standard for common warehouse meta-data interchange and to provide a generic mechanism that can be used to transfer a wide variety of warehouse meta-data. The objective is to define a rich set of warehouse models to facilitate the sharing of meta-data, to adopt open APIs) for direct tool access to meta-data repositories, and to adopt XML and XMI as the standard mechanisms for exchanging meta-data between tools. CWMI utilizes four key OMG industry standards for exchanging and managing object programming, design, and meta-data information:

1. XML - eXtensible Markup Language, a Worldwide Web Consortium (W3C) standard
2. MOF - Meta Object Facility, an OMG metamodeling and meta-data repository standard
3. UML - Unified Modeling Language, an OMG modeling standard
4. XMI -the XML Metadata Interchange specification, an OMG standard.

XMI is designed to use the Internet to exchange data between disparate platforms, supporting the needs of distributed development teams by providing a common framework for communicating design and data information. The advantage of the standards efforts is their broad vendor support. By choosing to align with a particular standard, IT is able to gain the opportunity of working with a wider variety of vendors, and is less dependent on the vendors being able to develop proprietary communications and integration methods between their products.

3.5.8 Vendor Management

In designing an enterprise Data Warehousing Infrastructure a single-vendor (SV) or multi-vendor interoperable (MV) approach can be employed. SV solutions are attractive because they offer the promise of one vendor being able to provide the solution and support it in the long term. In a SV solution the vendor provides all components of a solution and is responsible for maintaining and enforcing the interoperability of these components. Unfortunately, no one vendor has the breadth and depth to solve every problem to the satisfaction of every customer. An SV approach oftentimes forces the organization into a lowest common denominator solution, with excellent facilities for part of the system, and adequate facilities for the rest. The weak link in the solution will constrain the overall delivery. Working with a single vendor may result in a quicker initial solution, but leaves open the possibility of future disappointments as users become skilled with the technologies and their demands increase.

MV solutions facilitate the building of integrated solutions employing the best technology or set of technologies and have the potential of being better than any single solution provided by a vendor.

A MV approach enables the organization to work with the vendors that it prefers, the solution deployed is the best for the overall organization and its set of requirements, and has control over the enterprise data warehouse infrastructure. This is best suited for organizations developing custom client/server solutions However standards need to exist to enable the selected components to communicate with each other and to exchange data. The organization also needs to understand the requirements for each component, and be able to identify or build the components necessary for its solution. Also the organization needs to have the skills to integrate and maintain the overall environment.

Summary

Data Warehousing can provide significant new benefits to an organization. The technology has already attained the maturity phase. A careful consideration of the issues described in this chapter one should be able to reduce risk associated with implementing data warehouse solutions and deliver higher value solutions in a reduced timeframe.

Chapter Objectives

The objectives of this chapter are:

To highlight and discuss the implementation strategies for designing, building and deploying Enterprise Data Warehousing Systems.

Chapter 4

Data Warehouse Implementation

4.1 Introduction

The role of information in creating competitive advantage for businesses and other enterprises has been well documented. Whoever controls critical information can leverage that knowledge for profitability. The difficulties associated with dealing with the volumes of data produced in businesses brought about the concept of information architecture, which has spawned projects such as Operational Data Stores (ODS), Data Warehousing and Data Marts. Along with these came a set of associated complementary technologies which help companies collect, scrub, process, analyze and deliver useful information from this mass of raw, unconnected data.

Companies that focus on information find success when they can identify the usefulness of information, the source of that information and the appropriate location within the infrastructure of the organization that can profitably exploit that information. This entails offering data across organizational and application system boundaries by exploiting solutions that bring about enterprise-wide functionality. As corporations strive for efficiency, and begin to use Data Warehousing to share data across applications and organizational units, all of these issues come together into a central overriding theme: the concept of data quality. If one is to rely on the information in organizations to take action in business terms adequate care should be taken to make sure that the data upon which mission critical decisions are made is accurate, complete, and relevant.

Data Warehousing can help management and decision-makers transform raw data into useful information for planning and management. Today's global market place requires organizations to reach far beyond their traditional boundaries to secure information that ensures a competitive advantage. In addition, the contemporary approach to ensuring data quality usually begins with either a prototype solution that handles only the most rudimentary problems with limited quantities of data, or the acquisition of a solution in response to a crises within an organization to keep a larger warehouse project on track. These criteria serves as means to establish an objective standard for how well a system supports the dimensional view of data warehousing, and to set a target for improving their systems.

The following issues, which are classified under the respective headings, need to be considered during the planning for the implementation of a Data Warehousing system

i) Data Warehouse Architecture

a) Explicit declaration

b) Conformed dimensions and facts

c) Dimensional integrity

d) Open aggregate navigation

e) Dimensional symmetry

f) Dimensional scalability

g) Sparsity tolerance

ii) Administration

Graceful modification

Dimensional replication

Changed dimension notification

Surrogate key administration

International consistency

iii) Expression

Multiple-dimension hierarchies

Ragged-dimension hierarchies

Multiple-valued dimensions

Slowly changing dimensions

Roles of a dimension

Hot-swappable dimensions

On-the-fly fact range dimensions

On-the-fly behavior dimensions

The final eight expression criteria are analytic capabilities that are needed in common real-life situations. The end-user community experiences all expression criteria directly. These expression criteria for dimensional systems are not the

only features users look for in a data warehouse, but they are all capabilities that we need to exploit the power of a dimensional system.

Following are the expression criteria that need to be considered for dimensional systems:

Multiple-dimension hierarchies

The system allows a single dimension to contain multiple independent hierarchies. No practical limit exists to the number of hierarchies in a single dimension. Hierarchies may be complete (encompassing all the members of a dimension) or partial (encompassing only a select subset of the members of a dimension). Two hierarchies do necessarily have common levels or common attributes (fields), and may have different numbers of levels. Two hierarchies may also share one or more common levels but otherwise have no correlation.

Ragged-dimension hierarchies

The system allows dimension hierarchies of indeterminate depth, such as organization charts and parts explosions, where records in the dimension can play the roles of parents as well as children. Using this terminology, a parent may have any number of children, and these children may have other children, to an arbitrary depth limited only by the number of records in the dimension. A child may have multiple parents, where these parents' "total

ownership" of the child is explicitly represented and adds up to 100 percent. With a single command the system must be able to summarize a numeric measure from a fact table (or cube) on a ragged hierarchy for all members:

a) Starting with a specified parent and descending to all the lowest possible levels summarizing all intermediate levels

b) Starting with a specified parent and summarizing only children exactly *n* levels down from the parent or *n* levels up from the lowest child of any branch of the hierarchy, where *n* is equal to or greater than zero

c) Starting with a specified child and summarizing all the parents from that child to the supreme parent in that child's hierarchy

d) Starting with a specified child and summarizing all the parents exactly *n* levels upward in the hierarchy from that child

e) Starting with a specified child and summarizing only that child's unique supreme parent. A given ragged-dimension hierarchy may contain an arbitrary number of independent families (independent supreme parents with no common children). Conversely, independent supreme parents may share some children as I stated previously when discussing of total ownership.

Multiple-valued dimensions

A single atomic measure in a fact table (or cube) may have multiple members from a dimension associated with that measure. If more than one member from a dimension is associated with a measure, then an explicit allocation factor is provided that *optionally* lets the numeric

measure spread across the dimension's associated members. In such a case, the allocation factors for a given atomic measure and a given multivalued dimension must add up to 100 percent.

Slowly changing dimensions

The system must explicitly support the three basic types of slowly changing dimensions:

- Type 1, where a changed dimension attribute is overwritten
- Type 2, where a changed dimension attribute causes a new dimension member to be created
- Type 3, where a changed dimension attribute causes an alternate attribute to be created

This ensures that both the old and new values of the attribute are simultaneously accessible in the same dimension member record. Support for slowly changing dimensions must be system wide, as the following requirements imply:

a) Changes to a dimension that invalidate any physically stored aggregate must automatically disqualify that aggregate from use.

b) A Type 2 change must trigger the automatic assignment of a new surrogate key for the new dimension member, and that key must apply for all concurrent fact records loaded into the system. In other words, the creation of a new Type 2 dimension member must automatically link to the associated concurrent facts without the user or application developer needing to bookkeep beginning and ending effective dates.

c) the system supports ragged-hierarchy dimensions and/or multiple-valued dimensions, then these types of dimensions must support all three types of slowly changing dimensions.

Roles of a dimension

A single dimension must be associative with a set of facts via multiple roles. For instance, a set of facts may have several independent timestamps that you can simultaneously apply to the facts. In this case, a single underlying time dimension must be able to attach to these facts multiple times, where each instance is semantically independent. A given set of facts may have several different kinds of dimensions, each playing multiple roles.

v) Hot-swappable dimensions

The system must allow an alternate instance of a dimension to swap in at query time. For example, if two clients of an investment firm wish to view the same stock market data through their own proprietary “stock ticker” dimensions, then the two clients must be able to use their versions of the dimension at query time, without requiring the fundamental fact table (or cube) of stock market facts to duplicate. Another example of this capability would let a bank attach an extended account dimension to a specific query if the user restricts the query to a cluster of accounts of the same type.

On-the-fly fact range dimensions

The system provides direct support for dynamic value banding queries on numeric measures in a fact table (or cube). In other words, at query time the user can specify a set of value ranges and use these ranges as the grouping criteria in a query. All the normal summarizing functions (count, sum, min, max, and average) can apply within each group. The sizes of the value bands needn't be equal.

On-the-fly behavior dimensions

The system supports constraining a dimension via a simple list of that dimension. For the sake of vocabulary, call such a list of members a "behavior dimension." The support of behavior dimensions must be system wide, as the following requirements imply:

a) A behavior dimension can be captured from a report showing on the user's screen from:

- A list of keys or attributes appearing in a file extracted from a production source
- Directly from a constraint specification
- From a union, intersection
- Set difference of other behavior dimensions.

b) A user may have a library of many behavior dimensions and can attach a behavior dimension to a fact table (or cube) at query time.

c) The use of a behavior dimension in a query restricts the fact table (or cube) to the members in the study but in no way otherwise limits the ability to select and constrain attributes of any regular dimension, including the one the behavior dimension affects directly.

d) A behavior dimension may be of unlimited size.

e) A behavior dimension may have an optional date-stamp associated with each element of the list in such a way that two behavior dimensions can merge so that membership in the combined behavior dimension requires a specific time ordering.

4.2 Enterprise Data Quality Management

Enterprise Data Quality Management (EDQM), is intended to ensure the accuracy, timeliness, relevance, and consistency of data throughout an organization, or multiple business units within an organization, and therefore to ensure that decisions are made on consistent and accurate information.

Clean, useful, and accurate data translate directly to the bottom line for most companies. It represents the added revenues that are realized when businesses correctly model and track their customer relationships, product, or service preferences. Information is of value only if it is accurate, and in today's more complex information technology, when internal and external data are blended together in data warehouses and more advanced OLAP (on-line analytical processing) applications, new technology processes to ensure the accuracy of information are required. It is imperative to tackle the data quality issue from a point of prevention as well as cleansing existing data stores.

.

Data reengineering requirements are worldwide in nature. As organizations reach out to capitalize in the global market place, processes must be in place to support a variety of international data and values. While there is a need to establish worldwide customer identification and information standardization, there are few references available to validate international data elements such as corporate name, title, products, and services. These are some of the reasons to ensure an enterprise-wide standard of data quality.

Effective EDQM approaches can significantly lower the costs of data cleansing. The costs of data quality impacts organizations when existing systems fail to provide the data in the format necessary to profitably conduct business and results in scrap and rework remedies. Additionally, costs occur during assessment or the inspection phase of the process. Creating new business processes and standards throughout an entire organization can be an extremely difficult and costly process, as many companies discovered in implementing Total Quality Management programs on an enterprise-wide basis. Attacking issues on a business unit by business unit basis, using a consistent set of standards, allowed organizations to achieve their goals with more reasonable resource allocations, with the benefits of experience and cultural change from successful units assisting the transition to an enterprise process.

Similarly, addressing the issue of enterprise-wide data quality can be a massive undertaking that could overwhelm many departments. Companies that begin designing processes with the entire enterprise in mind start with a definable project and build from initial success, achieve consistent data improvement as they expand data quality through additional applications and organizational units. Developing programs to convert data from one format to another is not difficult. Designing processes to clean and standardize data on an enterprise-wide scale,

including data values that may not be obvious, presents a greater challenge. Fortunately, today's new generation of data management solutions provide data re-engineering and process tools along with conversion programs, to assist companies in implementing EDQM programs.

4.3 Data Quality and New Business Strategies

As a result of the success experienced with initial implementations of data quality software, organizations now understand the benefit of data quality as an asset in their business strategies. Traditionally, the initial implementation of a data quality process centered on the development of a data warehouse or large-scale system consolidation. Recently however, the trend has moved to the support of operational business strategies with leading technologies that enhance and add value to the organization. 2nd generation data quality software is one of these technologies. One common theme for each of these initiatives is a better understanding of and identification of existing and potential customers. Customer Relationship Management (CRM) solutions promise to help organizations know their customers with a suite of tools and processes that are designed to identify and link every touch-point that a consumer has within the organization. E-Business solutions, driven by the enormous increase in Internet activity, have identified an entirely new channel of product and service distribution as well as opening a direct communication link between the buyer and the seller. Lastly, Enterprise Resource Planning (ERP) suites provide companies with business solutions that integrate business unit software applications from financial and human resources to manufacturing and sales and distribution. This integration enables companies to optimize supply chains, strengthen customer relationships, and make more accurate management decisions.

Underlying each of these important business strategies in the need for high quality data. It is important to understand how data quality effects each of these business strategies and how your selection of an information quality solution may effect your organization moving forward.

4.4 E-Business and Data Quality

E-commerce allows organizations to deliver a personalized, high quality sales experience to customers, suppliers, distributors, and resellers providing more in-depth information and product availability. In the process, customers can make decisions more quickly and intelligently. The integration of each business unit with the necessary access to on-line customer information requires a data quality solution that is capable of identifying and consolidating customer information from numerous sources, cleansing and formatting the data, and matching and readying it for transfer to other end user applications throughout the enterprise.

E-business data cleansing and relationship matching requires a flexible solution set that allows one to leverage existing business processes. It should establish an enterprise standard for customer and product information quality. This process should incorporate an open, standards based computing environment, callable from the systems currently in place. The solution developed should be capable of integrating with all the existing platforms and applications. Data quality management processes are vital components to any e-commerce solution. In fact, data cleansing, validation and relationship matching functions should be integrated into e-commerce point of entry systems. Cleansing and reengineering data at an organizations' point of entry is important for several reasons:

> The speed with which data moves and the sheer volume of transactions occurring over the Internet has introduced a entirely new channel of

distribution for customers, resellers, and distributors. This new channel is a highly uncontrolled source of data and information.

Many different processes and users manipulate the data as it moves from the point of collection to end user applications that create data anomalies.

The need for the different data points at disparate business units giving rise to data inconsistencies.

The specific functions that need to be integrated in to the process are:

The need to standardize differences in data types.

Consolidate data from multiple and disparate sources.

Recognize and cleanse incorrect information.

Link and match related records.

Recondition data in free form text fields.

Geocode records for improved relationship matching.

Numerous other detailed data cleansing and transformation tasks.

4.5 Customer Relationship Management

One of the most exciting developments in the 1990's has been the ability to develop Customer Relationship Management (CRM) systems leveraging the abundance of economic, demographic, lifestyle, psycho graphic, DSS and Internet data available. The past two years have seen an enormous growth in product offerings on the Internet, as well as an increase in "families of products" as a result of mergers and acquisitions within more traditional institutions. Additionally, many companies have entered into complex sales and development

partnerships that has only served to increase the amount of customer and service data being housed within IT infrastructures. At the core of these efforts is the need to establish a clear view of the customer, be it individual, household, business or any combination thereof. Understanding the entire customer relationship and all of the "touch points" with customers, enables institutions to set about acquiring new customers, offer timely incentives for increasing business with established customers, and plan for long term relationships with the most profitable customers.

CRM is a process that incorporates a set of technologies that must be repeatable and consistent in order to facilitate the numerous touch points that customers engage in on a daily basis. The system in place at each touch point must be capable of identifying and accurately matching customer records in order to integrate the data from and to multiple sources.

Data from varied sources within organizations differ in format, degree of completeness, accuracy and standardization (i.e. last name, first name, MI). Additionally, there are also enormous complexities inherent in the data types, both customer and generic (i.e. phone numbers, SSN, product id's, codes, email, etc.) that aid in the identification of relationships between customers, households and businesses. It is clear that the success of any CRM program is linked and enormously dependent on the quality of the data and the ability of the organization to share consistent, accurate information across the enterprise.

Organizations require on a data cleansing solution that establishes standards that are portable, with output that can be consistently applied throughout the enterprise. Effective data quality is the result of an architecture designed to help solve the complex problems inherent in legacy system data, while also serving as

the operational gatekeeper for incoming data. For instance, e-business data cleansing and matching needs to be integrated with front end and CRM systems like Sales Force Automation (SFA), customer call centers and entry order systems. This provides customers with the ability to rapidly implement long lasting, sustainable, data quality solutions that cleanse and recondition large volumes of global data from multiple sources.

4.6 ERP and Data Quality

Enterprise Resource Planning (ERP) suites promise the benefit of increased ROI based on the integration of multiple applications along different lines of business. When data is changed or added, in the case of new customer or product information, each application in the suite is changed accordingly. This is a highly complex process that relies on sophisticated mapping of data within each application. Very often data must be transferred from legacy formats to formats required by the ERP system. When incoming poor quality data resides in those field formats, the outcome can be disastrous. Second generation data quality software solutions provide a seamless interface with ERP application suites for large batch file conversions and individual record and transaction cleansing. This ensures the quality and format of the data being migrated from one source system to another. As a result of the replication that usually occurs in ERP migrations an error occurring in one instance may be spread throughout the enterprise. Therefore, data should be cleansed as close to the source or entry point as possible. It is imperative then to cleanse the data prior to migrating to an ERP. Cleansing and standardizing at this point in the ERP migration is significantly less time consuming and less expensive and offers the benefit of ensuring the accuracy and consistency of the data from the outset.

4.7 Data Warehousing Trends - The Increasing Value of EDQM

Businesses have accepted as a given the value of Data Warehousing as a means to get their corporate data house in order. The size of the market, the speed by which it has achieved this mark, and the household names of corporations using it on a daily basis confirms this beyond dispute. Initially, the concept was to build a single, centralized, enterprise-wide repository, which combined all the data from all legacy systems and theoretically gave all users access to appropriate information. Some of these efforts were in fact successful, but most were extremely costly, laborious, time consuming, and political. One of the major issues, in fact, in deploying such a enormous warehouse is the data and metadata issue that arise from

multiple legacy systems all contributing data in different formats and standards with different contextual

meanings.

The industry responded to these issues by reengineering Data Warehouses and developing the concept of a more tactically focused warehouse known as a Data Mart. In this construct, multiple data marts are spread out through the organization and are used by individual or perhaps groups of departments to analyze their aspects of the business. This approach helped make warehouse technology more affordable and less time consuming. With multiple systems utilizing the same data, significant efficiency is gained. But this architecture has done little for the data quality issue. However this leads to multiple departments accessing the same legacy systems and trying to interpret the legacy data from multiple contexts simultaneously. Also an additional risk of a mistake in a local departmental data mart affecting company wide applications with potentially

significant negative ramifications. By bringing data cleansing to a system level, as a basic utility for the enterprise, significant efficiencies can be gained.

The best way to illustrate the value of data quality is to review the roots of bad data, and some of the ways corrupt data can impact an organization.

4.7.1 Errors

The origin of mistakes in data is the simplest of problems to understand. These include misspellings, typographical errors, out of range values, or incorrect data types. While typographical errors are difficult to correct, validation routines typically handle out of range values or incorrect data types within applications. An example of out of range values might be 13 in a field for month, or an alphabetical character in a numeric field, such as interest rate. Even if programs lacked validation routines, it is a relatively simple task to generate conditional logic to identify problem data. For example, if a product was introduced in 1998, code could be generated to reject any transactions dated 1997 or earlier. Nonetheless, horror stories abound, such as the Commonwealth of Massachusetts Division of Consumer Affairs, which reported 17 different spellings of "Boston" in its databases!

4.7.2 Homonyms

The English language contains many words and abbreviations with identical spellings that have multiple, and often unrelated or conflicting meanings, and relies on the context of usage to determine the correct meaning. Improper

interpretation of the context in which the homonym was used can have a significant impact on data accuracy.

For instance, a field entry of "No" may have several distinct meanings as shown below:

- The negative form of a reply, the opposite of yes.
- An abbreviation for the direction, North.
- An abbreviation for number, as in 1or 2.

Similarly, the contextual use of St. in the example below illustrates how context sensitive processing is built into our language, and that proper interpretation of data requires recognition of the format and condition of how words and abbreviations are used.

- Catherine B. St. James, MD
- In trust for
- Mary Church
- St. Catherine's Church
- MS ST 225
- 111 1st St.
- St. Petersburg, FL 33708

The use of St. is recognized in several different ways:

- As part of a last name in St. James.
- As a business name in St. Catherine's.

- As a mail stop station in ST 225.
- As an abbreviation for first in 1st.
- As an abbreviation for street.
- As part of a city name in St. Petersburg.

The complication of homonyms and the opportunity for misinterpretation of out-of-context data provides another potential source of data contamination.

4.7.3 Lack of Standards

When data entry responsibilities are spread among different people and business units, variations are bound to arise, as in the following example from information gathered for inventory purposes. Within a field as simple as Product and Location, data may be represented in several different ways.

- Product Location
- PC Bin Location 223
- Personal Computer Shelf 223
- Laptop On the shelf
- Notebook Yes

Abbreviations, sequence, and in many cases, the choice of fields for entering data can easily fluctuate from one business organization or employee to the next. The differences can be as trivial as the inclusion or omission of periods at the end of abbreviations or inconsistent capitalization. For customer records, the use of full names, nicknames, or middle initials can significantly add to the variability of data across different departments.

4.7.4 Legal Entities

In many instances, the addition or subtraction of naming conventions may alter the actual legal definition of a document. Many banking and financial institutions require complex naming conventions that are unrecognizable by most applications, but must remain intact in order to protect the legal purity of the document.

4.7.5 Missing/Invisible Data

Often data that is present may contain the proper structure and values, and in fact may appear to be correct but it contains data that has inadvertently been omitted causing identification and linkage mechanisms to unknowingly "grow" a mountain of poor quality data. This problem usually occurs without an organization's knowledge. For instance, "35 Avenue of the Americas" is syntactically correct. What are undetected are the thousands of apartments, suites and mail stops within the same address. Additionally, the name "Leslie Brown" is correct, but without a "title" deriving gender, matching would be accomplished with a lesser degree of certainty.

4.7.6 Phantom Data

In many applications, phony data (e.g., the date 99/99/99) may be used to flag a record or signify that there is no valid data for a particular field. Equally perplexing, the flag inserted into a field may have nothing to do with the data in that field; for instance, a phantom date may serve as an indicator that the record in question is no longer valid.

To be effective, EDQM as a process must meet a number of technical challenges. The process must work across multiple platforms and information architectures, must be adaptable and capture knowledge from an organization, and not scare away users by being difficult. When all of these challenges are met, EDQM can be leveraged into an Enterprise "Business Intelligence" Asset.

Interoperability is a critical technical challenge to EDQM. Today's enterprises not only contain a variety of computing technologies, from PCs to workstations to servers and mainframes, but also a number of database management systems, data architectures and applications with which EDQM must interface. Hardware flexibility is a critical component in choosing a data-reengineering tool. Adaptability and the ability to accumulate knowledge within an organization so as to facilitate reuse are critical factors for data-cleansing tools. Tools must be portable from one application to another, contain global functionality for world-wide data sets, be scalable for large as well as small applications, and have the ability to be re-used throughout an organization, building on prior knowledge and operational rules. Functional flexibility is another critical component in choosing a data-reengineering tool.

Ease of Use is critical in the successful implementation of EDQM within an organization. A successful data-cleansing application must be easy to implement, integrate with existing applications and business processes, and provide for monitoring and tuning of the system to ensure that knowledge and rules are easily maintained. Flexibility in use is another critical component in choosing a data-reengineering tool.

A data-cleansing tool with these three technical capabilities will facilitate deployment and consistent utilization of EDQM techniques throughout an organization. Once processes and procedures to ensure data qualities are in place, the organization can begin to leverage its data resources into a "business intelligence" asset.

4.8 Data-Cleansing

Once data warehousing architects and practitioners discovered the need for data quality, the next step was to find solutions to achieve it. Initially, data reengineering consisted of manually written code interposed between the data extraction and the data loading phases of the Data Warehousing implementation. Each project has specific needs, tailored to specific target and legacy data structures and context, and therefore each project required custom built edits to achieve the data quality required for the warehouse. Data cleansing has grown from this editing process in the early days of information systems through a series of first and second-generation tools to help manage data quality.

Proactive data quality initiatives start at the Data Entry phase. Data entry validation is the first line of defense against bad data, with validation routines checking data ranges and ensuring that all required fields are filled during the data entry process. Validation checks are commonplace in many newer systems.

Newer generation solutions often contain more sophisticated conditional logic that may narrow the range of acceptable data based on entries to previous fields.

The advantages of checking data quality at the data entry stage are fairly obvious: mistakes are nipped in the bud, while the information is still fresh, thereby avoiding the need for downstream rework that is often performed by someone unfamiliar with the source data. Data entry validations, however, are not necessarily foolproof. Just as a word processing spell checker will not catch grammatical errors with properly spelled words, data entry personnel can still input incorrect codes to the right fields in the correct format and range, and the error would go undetected. This is why the tools must be used in conjunction with enterprise standards, which allow certain accepted mechanisms for entering data and rejecting others. These must be implemented at an enterprise level in order to ensure that all departments involved in data entry (accounts receivable, order entry, sales) use the conventions consistently.

In many ways, Enterprise Data Quality is similar to Total Quality Management. The latter espouses the reliance on prevention of faults, while EDQM understands the nature of data inconsistencies and provides for reengineering at the source and within an organizations legacy infrastructure. Both espouse an enterprise-wide scope, and look to technology to augment and enable the successful implementation of the programs.

4.9 Enterprise Wide Solutions

The oldest data reengineering process is a manual process in which business analysts painstakingly review every record in a corporate database. This method

is quite labor intensive and inefficient, and its effectiveness is dependent on the quality of personnel, training and discipline. With corporate databases growing in size to hundreds of gigabytes, and in many cases multiple terabytes, manual approaches are no longer practical.

First generation data cleansing tools have several shortcomings. Most first generation tools are platform specific, which creates difficulties as systems migrate to updated hardware. Further, in an enterprise where multiple platforms exist (and this includes most of the organizations out there today), the utility of each of the first generation tools is limited to those platforms that the tool supports. This implied developing multiple cleansing tools, one for each platform, along with the concomitant issues of maintenance and changes due to changing business dynamics.

Many first generation tools are application specific, with cleansing logic embedded in code. This compounds the negative effects mentioned above for platform specific tools. When the cleansing logic is embedded in the code, we have a situation analogous to old COBOL programs where data and user interfaces were also buried in the code. We have learned a lot since then about the power of layered, modular software, and these principles now need to be applied to the first generation cleansing tools.

In addition, many first generation tools require the consulting assistance, and specific knowledge leaves with the consultants at the conclusion of projects. Again, given the dynamic nature of business today, this leaves the organization vulnerable to being dependent on consulting talent to maintain the currency of the system.

4.10 Enterprise Approach

Managing data quality throughout an organization requires an enterprise approach. Such an approach, which focuses on prevention and standards, as well as error correction, can provide significant benefits to users, information technologists and, most importantly, to the bottom line.Just as TQM focuses on the prevention of scrap and rework, EDQM focuses on ensuring the accuracy of data throughout the enterprise. EDQM requires changes to business processes and the development of standards, which ensure that data are entered and standardized in accordance with a set of rules, which adapt to changes in business needs.

4.10.1 Standard Processes

According to Deming, "True quality comes from the improvement of the process to eliminate defects, rather than from the inspection." Organizations strive to develop processes that ensure efficiency throughout their enterprise. It only makes "cents" to ensure the same quality with respect to their data. Larry English described the real cost to organizations in DM Review: "The costs of no quality result from the scrap and re-work of defective products and services and in customer lifetime value when non- quality causes one to miss a new opportunity or to lose a customer." English also believes that "Ensuring that a process is in place requires among other things, several key characteristics;

- The process must be defined
- It must be repeatable
- It has a process owner

- It results in reusable and reused data
- It captures data as close to the point of origin as is feasible
- It incorporates controlled evolution
- It involves developing the data model to support the major information views across the enterprise.

Few data entry processes are designed to ensure data quality. In particular, many older systems tried to save some time and effort by grouping all name information into one field. However, even a structure outlined with the above rules is not foolproof. For example, should the name of a product be written as "PC" or "Personal Computer"? One can observer that even this simple, valid variation between two alternatives will cause problems later on in merging data, and highlights the need for enterprise level standards which would avoid this situation. Of course, it is possible to write routines that will map these two variations of the spelling of the product name into a single consistent structure. But if there are a dozen data marts in an enterprise, does it really make sense to have each of the extraction and transformation processes for each of the data marts perform the same mappings on the same data over and over again? Or does it make more sense to perform this same edit check at the time of data entry, and enforce a corporate standard at the beginning of the process and save the effort of developing the code and reusing the code on a regular basis for multiple applications? The message is simple: don't keep cleaning the same data at the back end; fix it at the source. Lastly, it is imperative to ensure that the cleansed data satisfy the users and results in the ability of the users to make mission critical decisions.

4.10.2 Benefits

Besides the obvious benefits of avoiding the tedious data reengineering phases of data warehousing, there are numerous other benefits for the enterprise to adopting an Enterprise Wide Data Quality Initiative. First of all, we must remember that Data Warehousing is only one of many initiatives underway in large organizations today. Operational Data Stores, simple data queries and analyses into existing databases, and uploads and extracts into new applications are other areas in which the data quality issue rears its ugly head. "Garbage-in/garbage-out" works as well in a simple query into a sales database as it does into a galactic warehouse project, and the ramifications of a bad decision based on bad data are equally threatening. The need for clean data throughout the enterprise is universal. Figure 4.1 depicts the many locations within an organizations architectural infrastructure where an enterprise-wide data quality solution may operate.

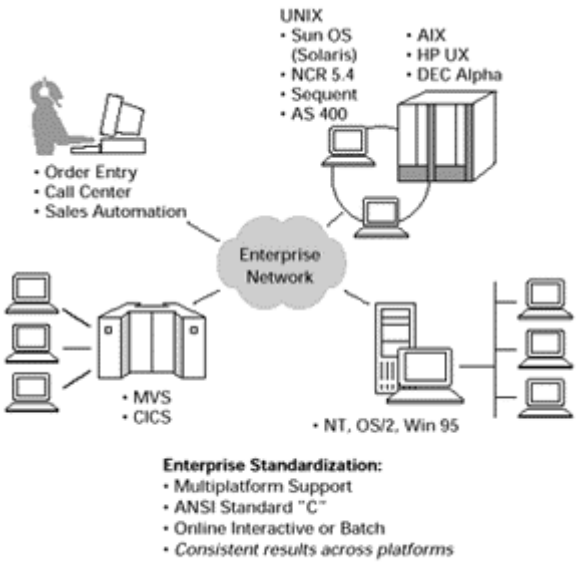


Figure 2. Enterprise Solution

Fig. 4.1 Enterprise Data Warehouse Systems [1]

The benefits of EDQM are also financial, and can be significant. In TQM, a rule of thumb is that prevention of an error is ten times less costly than fixing it. Similar rules of thumb can be applied to EDQM, with results between 10 and 100 times the cost of error correction, depending on the application. In direct marketing, the opportunity costs from incorrect mailings, even at relatively low response rates, can dwarf the costs of data cleansing. For example, if 20% of a 500,000 piece mailing were incorrect, the opportunity costs from the 100,000 inaccurate pieces can be staggering. Even if only 10% of the customers responded at $10 per response, the difference would be $100,000 in missed revenues from data errors. A small expenditure in EDQM can go a long way.

If an organization adopts a rules based approach to the concept of enterprise data quality, they gain additional benefits. By making the rules visible and overt, they accomplish two things: one, an enhanced, explicit understanding of what data exists in the enterprise, and two, simplified maintenance. Organizations are not static--they are dynamic entities, and the need to respond to changing business conditions means changes to the rules. As departments merge or break up, as divisions are acquired or divested, as products are added and deleted, the business rules associated with data-cleansing must also change. If we go back to our scenario of multiple data marts with multiple cleansing tools, each with the rules embedded in first generation tools or custom code, the changes must be implemented "n" times. An enterprise approach allows for the rule changes to be implemented once and then utilized "n" times--a far better scenario, and one which argues strongly for a corporate investment in Enterprise Data Quality. Additional benefits involve the automatic incorporation of clean data into any new systems to be implemented in the future.

4.11 Enterprise Data Warehousing Strategy

Business Intelligence solutions have earned their high investment levels because they have proven to be effective competitive weapons. Given the high investment levels demanded by BI systems, choosing an appropriate infrastructure strategy can make a significant difference where it matters most: your company's bottom-line financial results. Business Intelligence solutions help companies gain new insights into customers and markets, arming them to react more quickly and accurately to business opportunities. Increasingly these applications are being used to help companies design new ways of working with customers. For example, in consumer-oriented industries, such as financial services, BI solutions are being used to develop point-of-sale pricing systems that quote real-time prices based on the overall value of the customer to the organization, and the profitability of that specific transaction at a specific point in time.

In today's complex technical and business environment, IT has the challenge of developing a cost-effective business solution containing a robust data-warehousing infrastructure. IT must balance the need to develop an infrastructure that supports their current business requirements with the need to have scalability of the technologies they choose to implement as their business grows and expands. Additionally, IT faces the challenge of integrating infrastructure components from multiple industry vendors, or ensuring that the technologies they select can inter-operate with one another. Developing an overall enterprise data warehouse infrastructure strategy is key to successfully building and managing BI solutions.

4.11.1 Components

In an enterprise data warehouse, data is aggregated from sources that may span many divisions or organizations across your company. The sources are likely to

represent a full range of computing platforms -- from mainframe-based online transaction processing (OLTP) systems, to distributed departmental systems, to personal desktop systems, and possibly to include mobile devices. Not only are the sources scattered across multiple hardware platforms, but the sources also come in a variety of formats, from flat files, to spreadsheets, to hierarchical databases such as IMS, to relational databases possibly from multiple vendors. Large organizations can have tens, or even hundreds, of candidate source systems.

The first challenge to address is the identification of potential technologies that can be used for acquiring data from the variety of source systems you've identified. Data acquisition begins with finding the data that is needed for the data warehouse. Having located the data the next task is to work out a strategy to extract, cleanse, and transform the data.

Starting with extraction, it is possible to re-use the methods already in place to obtain data from an organizations operational database. There may already be some extraction jobs that have been written to replicate part of the data and share it among several applications. For example, a customer service help desk may take regular downloads from a master customer file. Management reports, which occur on a regular or ad hoc basis, may also provide some of the data. While not intended for this type of usage, database unload utilities can be a handy extraction mechanism. And finally, most database packages have import and export capabilities for exchanging data with other databases or applications.

Data quality and content also need to be one of the critical components that form a part of an organizations data warehousing strategy. The source data may contain inconsistencies, redundancies or other quality problems. In addition, frequently data will contain buried information that, if leveraged, could provide the competitive edge desired by an organization. Far too frequently, businesses fail

to capitalize on their investment in data because they are unable to identify and realize the value that is hidden in their operational data stores.

Data re-engineering technologies need to be included in the infrastructure to ensure that the highest potential data quality, content value, and structure is realized. These technologies facilitate the tasks necessary to accomplish these goals including source data investigation, standardization, matching and reconciliation, and re-engineering. The end result is that data will be appropriately aligned for Business Intelligence usage prior to loading it into your enterprise data warehouse.

Once the acquisition and cleansing challenges are addressed, methods for moving and loading data into the target data warehouse need to be formulated. Performance is a key load-time issue. Systems need to be available according to your negotiated user agreements. In order to achieve those commitments with gigabytes of data to be loaded into the data warehouse, special tactics are required to ensure minimal load, build, and indexing time. Native database bulk load facilities provide one mechanism, but they may require that the data be formatted, with flat files fully denormalized per your star schema design prior to loading, which may add to your overall development effort. Load tools that can generate native code using embedded database calls may provide the best chance at achieving the performance requirements.

Warehouse scheduling and management is also a key part of your data warehousing infrastructure strategy. The objective is to develop an operational environment for data warehousing that meets the organizational requirements for system availability and quality. Data warehouse scheduling will control the overall extract and load processes for an enterprise data warehouse and dependent data marts. Areas such as performance management and database administration also require a different approach than conventional OLTP

systems. Considerations should include special index creation, usage of aggregate summary tables, query monitoring, and meta-data management.

4.11.2 Positioning

A data-warehousing project needs to be positioned as technology to meet a business need. Positioning this way means that at the highest level, a business vision, aligned with the company's strategy, and supported by executive management, will ensure executive support for the significant capital investment required for data warehousing technologies. In order reach "technology to meet a business need", a business vision will need to be articulated. It is of prime importance that there is a shared vision that describes how the technology will help the business do something new that will make a difference.

At a tactical level, IT needs to develop strong ties to business users in order to understand their business needs and objectives. Coupling knowledge of business requirements with IT standards, systems topology, and operational requirements, IT can develop a conceptual system and data design. Strong business and technology skills are needed. Ultimately, the loop will need to be closed, with users being given a detailed description of the new system, with step-by-step discussions of how the new solution will help make the user more productive.

As usual, the right technical solution does not guarantee success. Enterprise projects that span business areas, like a typical data warehousing project, are by their nature risky. The implementation team needs to get both the technical and business sides of the project right in order to give itself the best chance for

success. Expert level tasks, such as technical architecture, meta-data modeling, data warehouse design, or data archaeology, require expert level resources.

The basic game plan for a well-rounded solution is:

- Pull together a multi-disciplinary team.
- Confirm the enterprise business vision.
- Establish enterprise data model.
- Select vendor(s) for technology and integration.
- Identify the scope of any interoperability standards, as well as the proprietary vendor extensions to the standard.
- Design and develop extract, transform and load routines able to accommodate a wide range of disparate source and target platforms.
- Design and develop the enterprise meta-data management system.
- Design and develop the warehouse management and scheduling approach test to prove and ensure that data that is moved through the architecture arrives at target systems in a timely and predictable fashion while meeting integrity, consistency, completeness, and granularity requirements.
- Use performance modeling and testing to ensure that load times meet availability requirements.
- Roll out the solution to users.
- Put in place the on-going procedures to keep the system up and running.

The detailed technical and functional requirements will depend on your organization, its situation, and its strategy. Candidate vendors, though, should address the following:

- Scalability, with both technologies and business models that enable the relationship to begin small and expand to enterprise-support.
- Data Acquisition and Loading facilities, with automated code generation as a way to minimize the amount of handcrafted system-level code.
- Meta-data interoperability with other vendor products, to ensure that meta-data definitions are broadly useable across the infrastructure and the enterprise.
- Warehouse management and scheduling, integrated across the infrastructure and extensible upstream and downstream from the infrastructure to control the full range of acquisition and loading requirements.
- Experienced professional services, to provide an integrated solution and the expert resources needed to help ensure success.

4.12 Considerations for a Successful Warehousing Project

Enterprises today, both nationally and globally, are perpetually seeking competitive advantages. It has

become an incontrovertible axiom that information is the key to determining how to gain such a competitive advantage. The problem today is how to deal with the mountains of raw data, which our ever more efficient information systems are

collecting, massaging, processing, deriving, and disseminating. Technology has provided the concept of Data Warehousing as one alternative to coping with the well-known information overload described above. Data Warehouses exist to help management and decision makers transform raw data into information and to help management identify key trends. It helps the enterprise foresee predictable events and act in anticipation of those events and helps management understand the entire picture of what has already happened. It allows them to develop a good systemic understanding of events, thus allowing focused reactions to those events, such as redefining and reengineering business processes to take advantage of that understanding.

A clear prerequisite to enabling management to accomplish all of the above is the fact that the data to be analyzed has to be accessible and flexible, and it has to be available in a format that is usable by the requester. To date, too much emphasis has been placed on the raw technology which embodies the concept of Data Warehousing, and not enough on the underlying and concomitant strategy, planning, business processes, and services which develop, maintain, and use the Data Warehouse technology.

Experience has shown that in projects where the technology has been perceived as a failure, the problem does not usually lie with the technology itself, but rather with the way in which the technology was applied. It is often applied to the wrong business problem, at the wrong scale, with insufficient training, and planning, and with little or no thought to how users need to access the data, etc. Some analysts' statistics show that over 50% of warehouse projects fail to meet their stated objectives. In order to mitigate the risks associated in these projects, the project must work in close concert with the business community that will benefit from the warehouse and also need to have a solid grounding from a financial return perspective.

Data Warehousing technology can benefit enterprises at different levels and scales of implementation. This applies from departmental systems running on commodity platforms such as Microsoft NT, and database servers such as DB2® Universal Database™ for Windows NT, on up to the enterprise level systems running on enhanced parallel MPP architectures such as IBM's SP/2 with DB2 Universal Database. Database Management Systems play key roles in the long-term viability of Data Warehouses. Issues such as easy access to operational data, scalability, and management of metadata (information about data in the warehouse) stand out. Therefore, thought needs to be given to what the initial strategy needs to be to ensure that an organization truly benefits from the technology. The proper strategy would be to deploy Warehouses such that they can grow and adapt to the changes that occur in the day to day environment including unforeseen ones. This calls for prudent planning in developing the strategy to select architectures, which can react flexibly to changes in market dynamics, organizational restructuring, economic fluctuations, etc.

4.12.1 Business Objectives

Data Warehousing is a rapidly maturing technology changing with every product release. There is no Rosetta Stone that will tell any one organization what works and what doesn't. Consequently, experience and evolution are the best overall planning principles, which can be deployed by management today. There are a number of criteria, which should be kept in mind as strategies are developed. The criteria can be grouped into three categories:

i) Business Criteria

ii) Process Criteria

iii) Technology Criteria.

4.12.1.1 Business Criteria

The first set of criteria have to do with the business problems at hand and what demands are made on the project by the business dynamics encountered that includes an understanding of the following:

- **Critical Success Factors**
- **Need**

No technology project will ever succeed if it is not properly aligned with the business Mission, Vision, and Goals. Therefore, it is vitally important to understand such elements of the strategy such as:

- What is the problem at hand? Is it a problem related to cycle time, customer satisfaction, more cost-effective decision making, better business intelligence, or a general lack of information with which to make decisions on any of the above?
- Which of the departmental or enterprise goals and responsibilities are directly related to the problem at hand?
- What are the Critical Success Factors - those things that must be done well to solve the problem?
- Which of the organizational components of the enterprise is best positioned to solve the problem?

- Which audience within this organization will best use this technology and how: executives, financial analysts, scientists, engineers, clerical and administrative users, line managers, others and why do they need this (who will it benefit)?
- How can this technology be used to solve the problem? This is where the alignment of the technology to the problem will occur.

4.12.1.1.2 Quantifying Benefits

Once the benefits of the technology have been aligned to the business objectives, they should then be quantified. The reason for this is that management must be able to answer the question: How will we know if this project is successful? The answer must fit the form: "This project will be successful if it allows us to achieve the following goals..."

In many organizations, quantifying benefits takes the form of a financial analysis, specifically a Return on Investment (ROI) analysis. A study by International Data Corporation (IDC), co-sponsored by IBM, showed that Data Warehousing could provide significant and impressive ROI numbers. The study, which included 62 participants, demonstrated that the overall ROI on warehouse projects was 401% with payback periods of two to three years.

What was interesting about the study, however was that the smaller, departmental implementations, sometimes known as Data Marts, had a 533% ROI, while the larger, enterprise efforts showed an impressive 322% ROI. This is as illustrated in the figure 3.2 below.

Fig 3.2 Data Warehouse Implementation ROI

The IDC study identified three kinds of benefits in the use of Data Warehouses.

Cost avoidance benefits. These benefits were the ability not to spend money that is presently spent on generating endless reports to end-users. This included the resources expended by IT to generate answers on ad hoc queries. In many ways, Data Warehousing represents a liberation for IT by providing users with the tools they have needed over the years to generate their own reports. By allowing the users to do so, one eliminates the often endless loops of the user requesting a report from IT, IT delivering the report, the user either not approving the report due to some miscommunication or changing their minds after seeing what they asked for, etc.

Efficiency gains from increased productivity among end-user professionals who gather and analyze data. The analyst who must stop an analysis to get information and who has to ask someone else to get the information loses efficiency in two ways. The first is in the loop described above, where there might be several iterations of request and response between the analyst and IT before the analyst is satisfied with the results of the request. The second is in the interruption of the analysis, and the inefficiencies associated with recovering the thought processes that were underway when the analysis was suspended.

Warehouse dependent savings due to decisions based on analysis that could only come from data in a warehouse. This is a quality-of-decision issue that comes from the fact that certain data associations may not exist in any one

operational system and therefore are not available to the analyst. By building those requisite associations and relations in the Warehouse, a situation where the whole is greater than the sum of its parts occurs and those relationships allow the analyst to do their work better, and become more effective. This is not a case of people making bad decisions prior to the advent of the system; this is about giving people better tools to empower them to do better work.

Not all benefit quantifiers are in terms of ROI. A recent article described how firms often eschew formal ROI analysis because they consider the data warehouse a strategic investment. In this case, these organizations were convinced prima facia that the benefits would be worth the costs. A cautionary word, however: make sure that there is an understanding of the projected costs before starting the project if there is no formal financial measurement technique such as ROI.

Regardless of the culture of the organization, whether it accepts soft benefits in its approval process or not, be sure to quantify the anticipated benefits in business terms. Without this, the organization will have no metrics to determine whether or not the project is successful.

4.12.1.1.3 Product Integration Issues

The current maturity level of the Data Warehousing industry leads to a proliferation of many disconnected offerings in the marketplace. Many small vendors have joined the fray with products targeted at one or two elements of the Data Warehousing architecture. As a result, there are very few offerings in the market, which answer all of the needs of a potential end user. This leads to a risk regarding how well different packages will integrate, which common platforms are supported, and so on. A trend is emerging where a number of large vendors form integrated product teams with smaller vendors that hold a solid solution in an

important warehousing niche. For example if a large vendor had a great DBMS solution, they might team with product vendors with extract technology, data cleansing technology or specialized OLAP tools. One benefit of these vendor consortiums is to agree on common approaches to sharing meta data between tools and databases from different vendors. Meta data, with its global significance, is a key to product integration in contemporary data warehousing solutions. The "investment" of these vendors integrating products from different vendors benefits end users by resolving the tricky and risky issues associated with integrating products from different vendors.

4.12.1.1.4 Analytical Applications

One of the fastest growing trends in the Data Warehousing market is the emergence of packaged analytical applications. The ability to script data analysis applications to derive particular end results is paramount to the evolution of effective data warehousing solutions. In this regard, early data warehousing solutions were all one-off customized applications built to deal with particular business challenges. This continues to be an important part of data warehousing activities. But in addition, as applications and business needs become more generalized and as custom applications begin to be widely deployed, there is a viable place for sets of off-the-shelf analysis applications that can be deployed quickly and effectively to meet many common business problems. These off-the-shelf applications also include the capacity for semi-customization and are the trend of the future. The costs of these applications should continue to drop, while the real value will continue to grow.

4.12.1.1.5 Cultural Issues

Data Warehousing is about pooling resources (data), which implies sharing, which in turn can imply a loss of control, a concept sometimes inimical to many data owners. This kind of organizational provincialism can sometimes throw up impediments to a Warehousing project and must be dealt with in the early phases of the project.

4.12.1.1.6 Business Unit and Process Considerations

Technology is only useful to the extent that it supports our ability to carry out our assignments and achieve our corporate goals. Therefore the introduction of any new technology must be aligned with the business units and processes, which it is intended to support. The IT organization may or may not have all the requisite technical skills, but IT will not successfully implement a warehouse project without the involvement and commitment of the business unit.

4.12.1.1.7 Technical Infrastructure

Very few, if any, IT organizations have the requisite combination of skills and resources required to perform all of the technology, planning and implementation tasks required for successful Data Warehouse projects. The chances that a single IT organization will have all of these talents available for this project at the same time are small. Therefore, at some point in time, many organizations will need to locate a partner to consult in the technical planning for the Warehouse and then eventually assist in the implementation of the system. The partner selected must be able to operate within the constraints enumerated in the organization's Warehousing strategy, including the methodology chosen, implementation style chosen (see Process Criteria - Methods), and so on.

Choosing the wrong partner, for example one who has Data Warehousing experience but not Data Mart experience, or one who does not have experience across the entire spectrum of products and services, can increase the risks associated with these systems. For example, IBM offers specific services in conjunction with its Visual Warehouse solution as well as custom services for any scope of warehouse implementation.

4.12.1.1.8 Standards

IBM, in conjunction with Oracle and Unisys, is sponsoring an OMG (Open Management Group) subcommittee for the standardization of Common Warehouse Meta Data. The objectives of this committee are to establish an industry standard for common warehouse meta data interchange and to provide a generic mechanism that can be used to transfer a wide variety of warehouse meta data. The intent is to define a rich set of warehouse models to facilitate the sharing of meta data, to adopt open API's (Java and Corba) for direct tool access to meta data repositories, and to adopt XML as the standard mechanism for exchanging meta data between tools. The subcommittee, chaired by IBM, is in the process of accepting vendor proposals for the above objectives.

4.12.1.1.8.1 OMG Committee for XML/XMI

A related OMG subcommittee has been formed to standardize XML Meta Data Interchange (XMI). IBM, Unisys and other industry leaders are also involved in this work. IBM and Unisys have submitted a proposal co-submitted by Oracle, DSTC, and Platinum Technology and supported by numerous other vendors. The proposal for an XML Meta Data Interchange Format specifies an open

information interchange model that is intended to give developers working with object technology the ability to easily interchange meta data between modeling tools and between tools and meta data repositories. In a data-warehousing context, the proposal defines a stream-based interchange format for exchanging instances of UML models.

One of the inherent risks of a new technology is the lack of standards, and Data Warehousing is no exception. There are countless examples of competing technologies that resolved themselves into one standard, and most of the time the resolution creates winners and losers.

4.12.1.2 Process Criteria

A number of elements of the Warehousing strategy have to do with processes: processes by which the strategy is implemented, and processes, which are supported by the overall strategy. Many large warehouse projects have failed because of an inability of the organization to handle the size and scope of the project. If an organization can come up with an integrated data model and solve all of the issues associated with such architecture, the benefits are indeed significant. However, this is sometimes not realistic and not necessary for the problem at hand. Industry studies have estimated the average size of a large warehouse project to be nearly two million dollars with a time to completion measured in years. This question of project scope should be readily answered from the exercise described earlier on aligning the technology to the corporate Vision and Mission. Once the questions relating to who needs the technology and which problems are being addressed are answered, the scope should be relatively straightforward to determine, which should allow management to allocate appropriate resources to manage it.

A strategy, which entails an enterprise wide scope, has certain implications associated with it, which should be understood. First and foremost it requires integration and cooperation among multiple organizational elements. Many issues will arise regarding different definitions of similar or identical terms, competing objectives and agendas, data parochialism and an unwillingness to give up control. This is a situation, which can be difficult to manage and successfully navigate. Oftentimes, change management is necessarily intertwined with this exercise, since the organization will have to wrestle with inter-departmental issues as described above. Management must assess whether or not the organization is ready to deal with these kinds of issues, or whether they might not better wait until a more appropriate time.

A Departmental, or Data Mart approach is by definition smaller in scope, more focused in its outcome, quicker to achieve, less costly. However, there is a risk to developing Data Marts in a vacuum, as described in more detail in the method section below. Ideally there is a need to think globally about future integration with other departmental applications and Data Marts to avoid developing Data Islands.

Data Marts have their place and present a strong business case for starting with such an implementation, but if an organization determines that an enterprise warehouse is the appropriate strategy then there are many successful models to emulate.

4.12.1.2.1 Implementation Approach Options

Deciding whether a Data Warehouse or Data Mart is right for the enterprise is an appropriate beginning. It must, however, be followed by a decision on whether to

buy an integrated package from a single vendor, engage a systems integrator to bring together a collection of best of breed products, or have the enterprise's IT department create a best of breed solution. The obvious advantage in dealing with an organization which can offer a complete solution is, of course, faster realization of business goals, usually at a reduced cost, and with a good degree of certainty that the ultimate solution will work (lower technological risk). These benefits do come at a price, though, and that price is the potential compromises that have to be made in functionality and performance by accepting the full suite of products from a single vendor. Further, the enterprise may have to adapt business processes to fit the specific characteristics of technology sourced from a single vendor.The best of breed concept is certainly not new. It has as its foundation the tenet that the enterprise will be better off if it can somehow bring together the best extraction tools, databases, SMP/MPP hardware, disk drives, analysis tools, network, etc. and get them to function as a unified system. Aside from the difficulty and religious wars that accompany the attempts to define best, the price paid for the anticipated exceptional performance is primarily the pain associated with integrating the disparate components. Different vendors value different architectures and functionality characteristics, and the best extraction tools, for example, may not integrate well with the best meta data repository, and so on. Data Warehousing meta data standards are still evolving, although IBM, in conjunction with Oracle and Unisys, is sponsoring an OMG (Open Management Group) subcommittee for the standardization of Common Warehouse Meta Data. This will result in a rich set of warehouse models to facilitate the sharing of meta data. Another OMG subcommittee has also been formed to standardize XML Meta Data Interchange (XMI). These standards initiatives will go a long way to resolving these important issues. Most integrators find that all projects require compromises to achieve integration, although some level of custom fit with the enterprise is generally achieved.

4.12.1.3 Technology Criteria

The technology dimension will of course play a major role in the enterprise strategy. Different strategies will require different technological characteristics and features. Just as the technology must align itself with the business mission, the strategy must also consider the technology.

Scalability

Scalability refers to the ability of a system to increase in capacity as users demand more, as data stores grow, as more users are added to the system and as more applications are developed against the Warehouse.

One of the challenges associated with introducing new technologies and new applications is that of determining the actual system load after implementation. JAD (Joint Application Development) sessions attempt to mitigate the risk by attempting to produce a picture of user requirements, but the truth is that users who have never had the opportunity to work with a new technology don't really know what to ask for. Therefore, as the users become familiar with the query capabilities and the navigational issues, their own success will cause them to demand more from the system as word spreads and more users exploit the features of the system. In time, users will become more sophisticated and begin exploring with ideas of data mining and visualization as their analyses become more complex. All of these are factors that demand scalability in a system. Companies acquiring warehousing technology need to be assured that scalability is "built-in" to the architecture.

Manageability

Data Warehouses require the development and implementation of new processes, tools, and work systems to manage the extraction/transformation of operational data, the administration of users (adding, deleting users, changing access rights) and so forth. In the course of deploying a Warehouse, a number of operational management decisions must be made and supported by the product solution set:

- What is the relationship and meaning of the data being loaded to the intended business use?
- How frequently should data loads and transformations be made?
- Should they be daily, weekly, monthly, quarterly?
- How will the Warehouse reporting adapt to the business processes as they change over time? How will the system model and track the business?
- How much data needs to come in on each load?
- How long will it take for the system to recompute all the indexes, meta data updates, and other administrative details that must be undertaken?
- How will the system monitor database operations?
- How will the system deal with backups/restores and what should the process be to administer a disaster recovery plan?
- If the system is to be paid for by usage chargebacks, is there a mechanism for the systems administrator to keep track of usage by

account number or password, and are there reports available to facilitate this feature?

Performance

How well the system performs will be the ultimate arbiter in the success or failure of the project. The intent of the Warehouse is to help people do their jobs more effectively and efficiently. If the response time is not adequate, users will not use the system, the enterprise will not derive any benefit from the expenditure, and the project may wind up with a negative ROI. Therefore the technology dimensions of performance must be thoroughly considered in developing a strategy.

Over the years, there have been a number of studies to determine the limits of human patience in dealing with computer response times. In general, users want to see something back in a timeframe that does not interrupt their thought processes. Some have said that two to five seconds is a good target response time. But when we are dealing with such gargantuan database sizes, it is difficult to conceive of doing a table scan on a multi-million row table in that timeframe, which leads us to the second human factors point. That is the fact that the amount of time a user will wait is proportional to the perceived difficulty of the procedure requested. Therefore, if a user knows they have entered a particularly nasty query, they will be more tolerant of delay. The best advice on this subject is to work with the user community in establishing meaningful metrics and working cooperatively to set and meet expectations on both sides.

There are several areas that directly impact the performance of a system: the hardware architecture and the database architecture. The hardware dimension is simply the use of Symmetric Multi Processing (SMP) and Massively Parallel Processing (MPP) architectures. The top end of the MPP capacity does outstrip the top end capabilities of SMP. However, the applications where this kind of performance is necessary are few and far between. For this reason, many industry analysts are predicting that SMP will be the hands down winner in the Warehousing market. This does not mean that every warehouse implementation should be on a parallel technology of one sort or another, many applications can perform satisfactorily on non-parallel systems.

Some database vendors have followed the hardware architecture by introducing parallelism into the database functions. For example, IBM has continually enhanced products such as DB2 and it now provides capabilities such as parallel query, loads, joins, scans, and utility processing providing tremendous value.

Flexibility

If there is one thing that distinguishes market conditions today it is the pace of change. In fact, as pointed out in the introduction, one of the drivers leading enterprises to consider Warehousing technology is the need to keep up with that change. Deployment of a Warehouse will not alter the pace of change or the need to keep up with the change, which means the Warehouse technology itself must be flexible, allowing for rapid responses to changing conditions. The Warehouse must be adaptable to changes in the enterprise, such as reorganizations, mergers, and acquisitions. It may be necessary to implement new queries based on responding to a competitors product introduction--queries not envisioned in the original design. If the database needs to be redesigned,

reloaded or indexed, this could mean a significant delay in responding to the competition.

Ease/Speed of Implementation

The development and maintenance tools available with the Warehouse will be key to the success of the project.

- Are the tool sets tightly integrated, as is usually the case in a one-stop-shop solution, or will the development team have to wrestle with the tools as well as with the disparate products to make them work together?
- Are the user interfaces graphical or the old command line metaphor?
- Do the development tools automatically generate the meta data content, or will there be a second step to build the meta data and then a third to reconcile the mistakes made in doing this manually?

These considerations apply not only to the initial deployment of the Warehouse, but also speak to the flexibility, since changes to the Warehouse will likely at some point involve the use of these tools again.

Tool Integration

Integration deals with how well and how smoothly the different architectural components interact. Obviously these are subjective terms, and therefore there are degrees of integration in many different areas such as administrative interfaces, Warehouse management, problem determination, etc. Tight metadata integration would yield benefits from needing only to input meta data once and, once operational, preventing meta data from becoming out of synchronization. For example, an extraction tool and a meta data repository could interact according to several different models as illustrated in the following figure 4.3.

Fig 4.3 Tool Integration Model –1 [1]

In the first model, each component has its own structures and syntax for the meta data and a third component is interposed between them to effect information transfer between them. In the figure 4.4 a translation module changes the formats of the information from one component to the other. Therefore, in this model there is no direct integration between components.

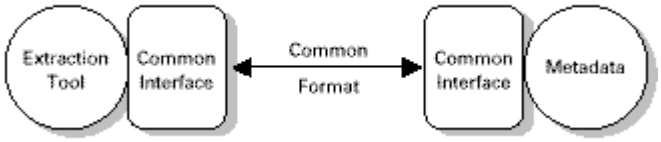


Fig 4.4 Tool Integration Model – 2 [1]

A second option is to have each module maintain different internal structures and syntax, but to have a common defined interface, such as a metadata standard. In this case, the accepted transfer mechanism and translation is built into each module. Compromises must still be made because the internal structural differences may yet interfere with some functions, but overall the interaction between the components is facilitated by the acceptance of a common mechanism for information interchange. This situation is considered to be moderately well integrated, and in fact do have an accepted architecture that each component follows to effect this integration.

Fig. 4.5 Tool Integration Model –3

Finally, on the other end of the spectrum are tools that use the same structures and syntax internally as well as present a common interface to each other as depicted in Figure 4.5. These components are completely integrated. The more vendors involved in the solution, the more challenges exist integrating them--some single vendor solutions make this option relatively achievable.

The degree of integration of the Warehouse is an important consideration in developing a architecture and an implementation strategy. First of all, the Warehouse must integrate with the existing architecture and infrastructure of the organization. If the Warehouse can only integrate with the existing architecture by means of extensive interposition of custom code, the project will be very

expensive, lengthy and complex. Questions must be asked about the degree to which the proposed Warehouse must conform to existing supported operating systems, networks, data standards, existing databases, existing application development environments, and more. If the Warehouse supports common, open architectures, such as in the second model, the likelihood of being able to add analysis functionality later, such as data mining for example, will be higher. Open architectures also make customization of products easier due to the more regular and predictable nature of the interaction among architectural components.

With regard to the existing operational applications from which the Warehouse will extract its information, the tighter the integration with all data sources, the better. Some of the systems in the enterprise will have flat files or older hierarchical or network database architectures which can present a challenge.

Data can be extracted from the databases in discrete batches or continuously, on a transactional basis. Depending on the Warehouse application, either of these may be preferable or both may be required.

The Warehouse should also integrate with enterprise standards, both de jure standards, those promulgated by officially sanctioned bodies such as ANSI, OMG, the Meta Data Council and de facto standards that are practices and products generally accepted in the industry.

Completeness

The completeness of a solution points to the existence of all the architectural components. This means that all of the parts an organization needs to work are

in place and functional, from extraction and transformation to storage and metadata, to the analytical, management, and change processes required.

Many large vendors, particularly IBM, have invested significantly in their core technology and have evaluated the issues surrounding Warehouse technology and implementation and developed partnerships and processes that address the issues discussed in this section.

Implementation Tactics

Following is a high level guideline for implementing successful Data Warehousing projects.

Planning the Project

> Many organizations are in such a hurry to install a system that they tend to gloss over this vital step.

Gaining Commitment at Top

> Senior Management Sponsorship is crucial in these projects, and the level of support is commensurate with the scope of the project. If the project is Departmental in scope, then the senior Departmental leadership must be on board.

i) Establishing a Team

In today's dynamic environment in which data warehousing solutions are becoming key ingredients to business success, users are learning nearly as much about their data requirements as OLTP users. Therefore, it is critical to form a team where both technical and end-user personnel can work together to develop a mutually acceptable and technologically achievable set of specifications and requirements. This should result in continuous collaboration throughout the project as many warehouse projects can be viewed as a discovery process where the business perspective must be weighed alongside the technical issues.

Determining Metrics for Success of Project

As part of the planning process it is critical that metrics be defined which clearly demonstrate that the results of the project meet and are aligned with the original business goals which drove the project to begin with. It is also imperative that the metrics be objective rather than subjective.

ii) Project Plan

A project plan detailing the objectives, approach, strategy, ownership, timeframe, and resources' responsibilities is a must if the project is to be managed with any degree of professionalism and efficiency.

iii) Identifying the Areas of Expertise

One of the outputs of a good plan is identification of the resources required and an analysis of whether they exist in-house and whether or not they are available. Make sure they are represented on the team. Having senior management support is a good prerequisite for being able to get the people you need, and if they are not available, for being able to get permission ahead of time to go outside (hire service providers) if necessary.

iv) Developing a Communications Plan

A Warehouse is only useful to the organization if users exploit its abilities. It is foolhardy to spend effort on developing such a project if the concerned department or enterprise is not ready to take advantage of it. A communications plan is essential to disseminate information about the project.

v) Using Benchmarking Techniques

If possible, investigate other organizations that have successfully completed projects similar to the one at hand. This can take the form of literary research or actual trips to view operational systems, interview users, developers and management as to the best practices they have encountered in their project.

vi) Selecting a Methodology

A methodology that is known and accepted by the organization will go a long way to smoothing out many of the rough spots which any project will hit. A methodology provides three components to a project:

A logical series of activities to achieve a desired end. This structure will tell the organization what has to be done and when in order to produce a quality product. A definition of deliverables associated with each activity and progress reports on the business benefits from each activity. This tells the organization what the output is of each step.

vii) Roles and responsibilities for actors in the activities.

The Methodology will also help define a project structure and what has to be done to manage the project: when reviews are to be held and what each review will cover. There has to be a blending of business and technology in order for these projects to succeed, and therefore the lead people have to be sensitive to the various cultural differences of the team members.

viii) Using Service Providers

Any organization will have areas where there is either a lack of expertise or where there are insufficient skilled resources available for the project. The project manager should look for areas of expertise, which are not represented on the team. With Warehouse tools such as OLAP, reports become three-dimensional and one has to pivot and drill through them searching for information. Designing these reports with little or no prior experience is very challenging in that you must think differently. Users will need support in learning, specifying and evolving the output of data warehouses.

ix) Architecting the Solution

Architecture is defined as the definition of the components of a solution and their interaction. Defining the components of the solution and how they interact is a critical step in implementing a successful project. The solution should be an end-to-end solution and allow for all the characteristics described earlier, including scalability, extensibility and manageability.

The choice available to IT range from tightly coupled and integrated product "suites" provided by some vendors (e.g., IBM's Visual Warehouse™) or individual/groups of building blocks of Warehouse products that IT would take the primarily role in integrating to meet their needs.

Defining a directory such as IBM's Visual Warehouse Information Catalog into the architecture will allow the system to draw business metadata from dozens of other products, including DB2, Oracle, Sybase, Hyperion Essbase, CASE tools, and more. Tools such as the Information Catalog can help provide users with the keystone component of consistent, synchronized metadata that helps developers, maintainers and users alike.

The design strategy is as illustrated in the figure 4.6 below.

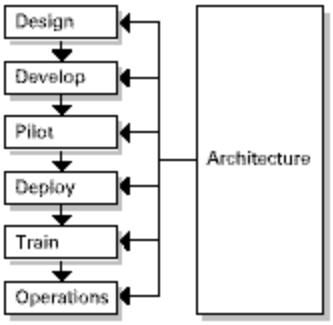


Fig. 4.6 Data Warehouse Design Strategy

x) Designing the System

Designs should be based on open, well-understood architectures with well-defined interfaces between the components. Use of standards will help the scalability and flexibility of the system over time. Navigational tools for users and user interfaces in general should be designed with the business problem in mind in a cooperative process with end users and managers.

Departmental systems should be designed within the context of an enterprise framework. That is, where possible and practical, identify data elements and concepts that span multiple departments and try to define them in the broadest possible terms so as to be able to include other departments later on.

xi) Developing the System

The development environment should be capable of using RAD techniques to help end users who might not be able to grasp Warehousing and analysis concepts immediately. One of the challenges associated with building a system is the ongoing integration of meta data. The value of integrated metadata is that it reduces the time to implement a Warehouse, as well as providing greater maintenance and management efficiencies. When metadata is updated manually this process can introduce increased error rates for end users (e.g. defining tables incorrectly). One solution is put forward by the Meta Data Council; a corollary to the OMG (Open Management Group) subcommittee for the standardization of Common Warehouse Meta Data and the related OMG subcommittee to standardize XMI. It is a federated approach whereby all stores of metadata type information (data dictionaries, passive repositories, encyclopedias, data base catalogs, etc.) pass through a "metahub" that changes the syntax and other relevant information about the data. This allows any tool that adopts this approach to interoperate and interchange metadata within and around the Warehouse on an ongoing basis. This approach is not yet widely available but holds great promise for resolving one of the major issues in meta management.

xii) Prototyping

Pilot systems, also known as prototype systems or proof of concept systems are among the most misunderstood concepts in the industry. There are three possible reasons for an organization to embark upon a pilot program:

- The technology is foreign to the organization, and there is a need to understand the benefits of the technology to see how it might help in the business problems at hand.

- The technology is understood, but finding an appropriate application is not. The organization wants to know if a particular application of this technology to a business problem is appropriate, which is to say, will this technology solve the problem?
- The technology is understood as well as the application, but the cost/benefit equation is not understood. The organization wants to know if the cost of the solution is worth it.

Each of these motivations will result in different pilots, in different places in the organization and with different associated metrics.

Regardless of the motivation, however, a primary question that must be asked is: How will we know if the pilot answered our questions? The only way is to develop quantitative as well as qualitative metrics and in addition develop tools and techniques for capturing and analyzing the results at the end of the pilot program.

xiii) Deploying the System

If the strategy calls for replicating successful projects in other departments, then a roll out plan must be developed to identify the order of the rollout as well as any integration efforts that must be dealt with. These might include process reengineering, especially if multiple departments are today involved in a process that is not automated, and one of these departments will receive the automation prior to the others.

xiv) Training Users

A equally important though neglected aspect of the design that complements a technically outstanding Warehouse is the training of the en\d users and the administrators of the system. Training programs must be developed which target not the technological niceties behind the screens, but rather that address the nuances of using the system from a business viewpoint. Focus on training the user on understanding the meta data and the navigation capabilities as well as how to use those in analyzing a business problem.

xv) Maintenance

Developing a Warehouse is one thing, keeping it operational is another. It requires managing the timeliness of file transfer and load processes as the system grows in size and complexity. Daily operations create meta data changes, such as the addition of a user, or loading of a new star schema, which means the meta data repository must be managed. In addition, the organization will need feedback reports on the Warehouse operations. Managers will want to know that the quarterly data extraction actually was started on schedule and completed on time with all appropriate indexes regenerated.

Some users, in spite of all best training efforts will construct queries that are capable of bringing a system to its knees. In these situations, organizations will need database management tools that report on how much system resources are being used by a query and allow the interruption and subsequent termination of the query.

Some reporting tools are also able to identify which areas of the database are being hit hardest, or most frequently, or even identify which times of the month/quarter/year those areas are most likely to be accessed. This will allow managers to govern the extraction and indexing processes--

change table structures, drop columns, define different aggregations--according to anticipated uses.

The figure 4.7 below represents the basic elements of a Data Warehouse architecture that responds to the needs of enterprises today. This model is slightly different from most of the ones being promulgated throughout the industry in that it combines the technological elements with the process and planning elements mentioned earlier. The human figures in the diagram represent those elements that are primarily of a service nature, such as planning, analysis, processes and management.

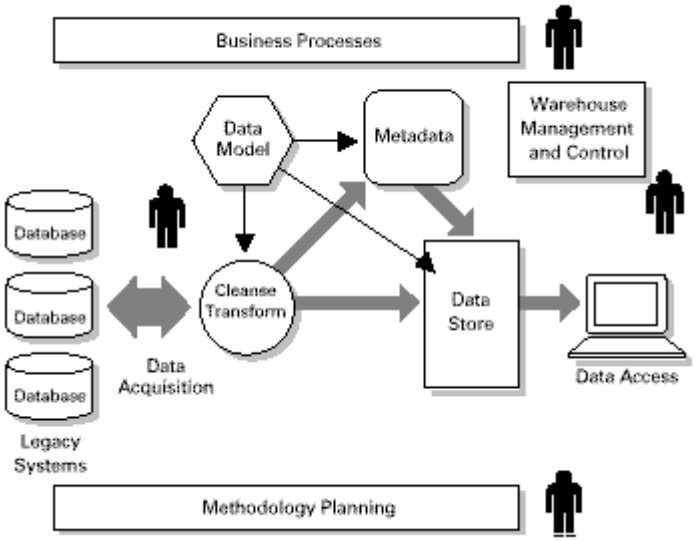


Fig. 4.7 Data Warehouse Architecture

Summary

Enterprise Data Quality Management is an emerging discipline, which has already taken root in leading edge companies that place a high value on information. These companies, aided by second-generation data reengineering solutions, have developed data standards and business processes, which ensure accurate data at its source. Ensuring the accuracy of the data up front pays big benefits in the back end, as wasteful cleansing and re-cleansing of the same data is avoided, not only in data warehousing projects, but in any other IT project involving data from multiple systems. Today's businesses are facing tremendous challenges as they seek to transform their companies into e-businesses, expand into new markets, which have been spawned by the reach of the Internet, and proactively manage their business for competitive advantage. Business Intelligence solutions have been successfully deployed by companies in many industries and have proven to be effective competitive weapons. Business Intelligence Solutions will continue to be one of the top IT investment areas for the foreseeable future. The demand for BI solutions is limited only by the need for better decisions (which seems insatiable) and the imagination of a given business' decision-makers.

The IT area is responsible for implementing the technologies of the Business Intelligence solution , including the data warehousing infrastructure. The challenge for IT is to define an architectural approach that meets the immediate needs of the company in the areas of functional requirements, cost, and risk, while best positioning the architecture to accommodate unanticipated future technical and business changes. Given the complex technical and business environment, the be st strategy to meet this requirement is to pursue a framework that supports a multi-vendor solution.

Increasingly, standards are being developed by major standards bodies to support interoperable solutions. Also, major vendors like IBM are recognizing that customers will demand multi-vendor solutions, which exploit best of breed technologies, and are responding by "stepping up to the bar". This means developing products that are standards-compliant, fostering alliances with complementary vendors to offer the best combined solution, and providing integration assistance to customers to help them implement the technologies needed for data warehousing in a controlled and cost-effective manner.

Chapter Objectives

The objectives of this chapter are:

To present an overview of the commonly available data warehousing tools and their features.

Highlight the organizational benefits of implementing data warehousing systems using real life case studies.

Chapter 5

Data Warehousing Tools & Case Studies

5.1 Introduction

It is estimated that the majority (75%) of the effort spent on building a data warehouse can be attributed to back-end issues, such as readying the data and transporting it into the datawarehouse. Data quality tools are used in data warehousing to ready the data and ensure that clean data populates the warehouse, thus enhancing usability of the warehouse. This research focuses on the problems in the data that are addressed by data quality tools. Specific questions of the data can elicit information that will determine which features of the data quality tools are appropriate in which circumstances. The objective of the effort is to develop a tool to support the identification of data quality issues and the selection of tools for addressing those issues. A secondary objective is to provide information on specific tools regarding price, platform, and unique features of the tool.

Attention to data quality is a critical issue in all areas of information resources management. A new airport in Hong Kong suffered catastrophic problems in baggage handling, flight information, and cargo transfer. The ramifications of the dirty data were felt throughout the airport. Flights took off without luggage, airport officials tracked flights with plastic pieces on magnetic boards, and airlines called confused ground staff on cellular phones to let them know where even more confused passengers could find their planes. The new airport had been depending on the central database to be accurate. When it wasn’t, the airport paid the price in terms of customer satisfaction and trust.

Data warehousing is emerging as the cornerstone of an organization’s information infrastructure. It is imperative that the issue of data quality be addressed if the data warehouse is to prove beneficial to an organization.

Corporations, government agencies and not-for-profit groups are all inundated with enormous amounts of data. The desire to use this data as a resource for the organization has increased the move towards data warehouses. This information has the potential to be used by an organization to generate

greater understanding of their customers, processes, and the organization itself. There potential to increase the usefulness of data by combining it with other data sources is great. But, if the underlying data is not accurate, any relationships found in the data

5.2 Data Quality Tools

Data quality tools generally fall into one of three categories:

i) Auditing

ii) Cleansing

iii) Migration.

Data auditing tools enhance the accuracy and correctness of the data at the source. These tools generally compare the data in the source database to a set of business rules. When using a source external to the organization, business rules can be determined by using data mining techniques to uncover patterns in the data.

5.3 Data-Cleansing Tools - First Generation

These first generation tools consist of batch processes that analyze the data for defects and construct routines for fixing the data as it parses from the source to the target system. Unfortunately, these processes, which rely on the paradigm of creating layers or streams of programmatic logic, have in the end, created complex, and unmanageable applications. This type of first generation product, which requires an enormous commitment of time and resources, cannot be easily maintained, moved or re-used, thus making them useless as enterprise solutions. Additionally, these tools tended to be retrospective, detecting and correcting

errors, rather than pro-active in preventing errors and correcting data through processes at its source.

5.4 Bulk Mail Formatting Tools

For many years, mass mailing houses have offered their own proprietary routines to reform name and address records to conform with bulk mail regulations of organizations such as the United States Postal Service. These services originated with the rental of mailing lists for direct mail campaigns.

Bulk mail formatting tools identify undeliverable addresses by performing matching to valid data lists. They identify exact duplicate addresses and format the information in a form that is required by the postal authorities for the purpose of improving postal delivery efficiencies. They lack the ability to create complex customer relationships (i.e., customers with more than simple name and address information on the label). Typically, they add bar coding and sort the mailing list by zip code to take advantage of favorable third class postage rates. While they generally perform well for processing simple address formats, they focus little on customer name data. Their goal is to scan for address delivery data and ignore all else. By design these tools process address information only and do not have the capability to reengineer generalized data.

Recently, commercial software products have packaged this capability. In general, these are batch-oriented, solutions, which use only a single set of rules that is difficult to modify. This "one size fits all" approach has limited utility in many applications. Furthermore, simply re-running these tools each time a mail list is scheduled adds no benefit to the quality of the data.

The advantage of these tools is that they require only modest data preparation effort on the part of the user organization--the tool performs virtually all of the work with limited user interface. The disadvantage, however, is that they are single purpose and perform only a mechanical function, and do not provide any benefits from clean data.

5.5 Vertical Industry Tools

Regulatory pressures have prompted the use of industry-specific solutions for industries such as banking and insurance. In general, these approaches reformat records using industry-standard terminology and file layouts. Furthermore, because banking and insurance firms often carry records of multiple members of a household, tools developed for these industries are effective for identifying family relationships.

The advantages of vertical industry approaches are the close fit to the business requirements of the target user (e.g., banking, insurance). However, because markets for these tools are limited, they are usually expensive. Furthermore, the logic is usually proprietary and difficult to modify, and often requires significant custom programming. Frequently, organizations rely on consultants and/or vendor provided professional services to provide the expertise and business logic that organizations need to make the tools work. Reliance on consultants inevitably requires the user organization to entrust business processes to outside experts, whose knowledge departs with them at the conclusion of the project.

5.6 Programming Tools

Programming tools rely on proprietary algorithms that can apply to almost any industry or data type. The prime drawback, however, is that they are code writing tools, not solutions. Each new cleansing problem requires additional code generation, which adds layer after layer of complexity and requires tremendous maintenance support in order to ensure success. These coding tools provide no inherent intelligence and must continuously be re-written for each new application. In most cases, they are also limited to specific platforms and databases (most are designed to operate only in a legacy environment). Users must either spend considerable time and effort learning the programming language and methodology, and their effective use as general-purpose tools often requires expensive consulting support. Because these tools require

extensive programming, projects typically extend over long timeframes, and, if consultants are instrumental in the process, the issue of the degree of knowledge transfer also appears.

Many companies have found that the use of first generation tools adds to the complexity of legacy systems and their operations. Typically, large companies end up with a variety of solutions ranging from internally derived to purchased, each with their own tools and techniques, which require training and support. In addition, the management of different data reengineering tools requires additional attention and resources, and migration problems between different tools for different applications. Thus, organizations find that they have many different data-cleansing solutions, all cleansing the same data in a different way. Not only is there the problem of maintaining and training for multiple tools, but there is also the potential problem of varying results on the same data, simply due to the vagaries of custom software.

A set of new generation data-reengineering tools has emerged which support Enterprise Data Quality Management. These tools differ significantly from first generation tools in that they are:

Versatile and powerful

Optimal solutions provide context sensitive processing for large volumes of generalized and name and address data both on-line at the point of entry, and in batch mode within legacy systems.

Portable and platform independent

In a typical environment in corporate America today, we find a multitude of platforms, from mainframe systems old enough to drink and vote, to mid-range systems such as UNIX, to lower end systems including NT and even PC-DOS based platforms. Data resides on all these platforms, and often in

order to develop a complete picture of the enterprise data holdings, extractions must be made from combinations of all of these.

i) Standards based

A corollary to the portable and platform independent, tools based on standards make the training and maintenance issues much more manageable, as well as helping to codify and standardize the entire environment around data quality.

ii) Global Functionality

As with all technologies, data cleansing and reengineering practices must focus on the global market-place. The requirements for processing international customer data present enormous challenges. For instance, there is a wide variation in data standards from different countries and the level of completeness of the information captured. The data often have many unique characteristics and conventions, some of which only have meaning in a local context. Second generation data cleansing solutions require the facilities to process international data, both generalized and customer focused. Ideally, these solutions provide both the core technology for tunable, user defined business logic, and pre-built intelligence that prevents organizations from "re-inventing the wheel."

iii) Extensible and Customizable

This means that the tools are flexible enough to adapt to different kinds of data environments and can handle a variety of rule sets. The rules based tools are also adaptable and reusable. Many are built on a Knowledge Base, which brings the benefit of many years' worth of experience and a broad range of data types to your application.

iv) Ease of Use

This means an intuitive approach to data cleansing. Use of highly customizable (tuning) through the use of text files as opposes to code generation. Built in intelligence that is portable, and easy to modify and enhance.

Because of all of this, second generation tools are better suited for enterprise approach that can be tailored to multiple data mart implementations on multiple platforms; can be retrofitted to OLTP applications also on multiple platforms to prevent data problems rather than fixing them later, and can be reconfigured more easily to respond to changing business conditions.

5.7 Complimentary Tools

Data extraction and metadata tools provide a second line of defense. These are generally deployed when migrating data to a new database or platform environment, and use tools to identify source data, transform data into the proper format, and move data to the target database. Data extraction and metadata tools provide very coarse refinement of data in that source data can be adjusted to fit the new system's format. Like data validation checks, automated transformations do not necessarily ensure that data are correct, logically consistent, and free of duplication.

5.8 Data Warehousing Case Studies

Case Study 1: BC TELECOM - Business Intelligence

Objective:

To reach the fullest potential and the greatest possible capacity in providing business intelligence from the corporate data warehouse.

Solution:

An existing, over-stressed NCR/Teradata data warehouse was replaced by IBM's higher capacity, more comprehensive data warehouse solution.

Need:

In 1993 when the Canadian telecommunications marketplace suddenly came alive with the introduction of long-distance competition. Then in January 1998 local competition came onto the scene. The marketplace became more dynamic and Canada became one of the most competitive markets in the world.

Business Intelligence from IBM

BC TELECOM not only anticipated this flurry of newcomers, it implemented a long-term strategy to quickly outflank them. To address this strategy and important related issues, business champions, IT professionals, experienced data warehouse and data architecture practitioners needed to work together to quickly identify goals, set implementation milestones and ensure deadlines were struck to.

Company Profile:

BC TELECOM is the second largest telecommunications company in Canada, providing a wide range of wired and wireless products and services to residential and business customers in British Columbia. It has one of the world's most advanced telecommunications networks, offering local and long-distance voice, data and image services, cellular, paging, and Internet service, content and access. The company takes an aggressive stance in earning and retaining its customers. The implementation of Business Intelligence tools and strategies has played a large part in formulating BC TELECOM's marketing tactics. The information gained from the IBM data warehouse has enabled us to understand

our customers across most of our companies," affirms Ed Michaels, and that's essential to gaining competitive advantage."

Data Warehouse - The Enabler

When BC TELECOM began to investigate the effectiveness of its data warehouse in the mid-1990s, the team already knew that getting the right information – information that had business value -- was the key to competitive success. The team considered their existing data warehouse a key "enabler." It would help move the company to a more customer-centric position, and it was essential in increasing revenues, reducing operating costs, and improving productivity and efficiency. The data warehouse could provide valuable business insight when managed properly.

By 1996, though, capacity in BC TELECOM's existing NCR/Teradata data warehouse environment was extremely stressed -- both in terms of processing capability and information storage capability. BC TELECOM learned that IBM had a network of existing large (more than one terabyte) data warehouse customers and further IBM agreed to a benchmark using BC TELECOM's data and queries. In that benchmark, scalability up to a terabyte was demonstrated. BC TELECOM's five essential components were:

i) Corporate Data Warehouse Management
ii) Data Acquisition Tools
iii) Data storage
iv) Data Access and Mining Environment
v) The Corporate Data Warehouse Directory.

Using the best-of-breed software tools, IBM's data warehouse solution provided BC TELECOM with all five essential components. IBM's existing relationships with development partners like Evolutionary Technologies International, Inc. (ETI) and Vality Technology, Inc., enhanced the overall package. ETIüExtract was used as the extract, transformation, movement and load tool, and Vality's

INTEGRITY was used as the data cleansing and data reengineering tool. Today, the relationships and the results from the use of these tools continue to be positive.

In October 1996, installation began. The team converted the old data warehouse business processes, adding much more breadth and depth to the base. They also created new business processes, adding much-needed business value. The new solution included an IBM SP2 complex with 56 nodes, a high-performance switch for internodal communications, and 3.7 terabytes of disk storage. The rollout of this system to users began in April of 1997; only seven months after the project began. Today, the user base has reached 75 and is growing. New business intelligence continues to be developed by knowledge workers in the business units. New data sources are being assessed and integrated. New business applications are being designed and developed. "IBM's solution has enhanced our ability to respond to competitive changes in the marketplace," relates Ed Michaels. "The solution has given BC TELECOM the capability to be significantly more agile in responding to our competitors' actions."

IBM's technology was a multimillion-dollar investment for BC TELECOM. The company initially anticipated it would be 18 months before seeing a return. However, soon after the rollout, the system had already proven its value.

The Future

BC TELECOM has chosen Information Advantage's DecisionSuite and WebOLAP to develop customer marketing and sales applications. The WebOLAP software provides the flexibility needed to handle the massive amount of detail data and diverse business requirements -- via BC TELECOM's Intranet. And relevant to IBM, a large SP complex that will contain both the SAP operations and the data warehouse has been purchased and is currently being implemented. In a blaze of deregulation and stiff competition, BC TELECOM's operating revenues still increased 9.4 percent in 1997. The new data

warehousing environment certainly played a part in that. The data warehouse provided he company the ability to really understand their customers and offer them exactly what they want

IBM's principal solution for generating and managing Data Warehouse and Data Mart systems is Visual Warehouse. Other IBM offerings such as the Data Replication Family and DataJoiner (for multi-vendor database access) can complement Visual Warehouse as data is moved from source to target systems. IBM also partners with companies such as Evolutionary Technologies International for more complex extract capabilities and Vality Technology Inc. for data cleansing technology. In addition, IBM has key partnering arrangements with Brio Technology, Business Objects, and Cognos for query and reporting as well as with Hyperion for OLAP technology.

For the warehouse database, IBM offers industry-leading DB2. The DB2 family spans Netfinity systems, AS/400 systems, RISC System/6000 hardware, IBM mainframes, non-IBM machines from Hewlett-Packard and Sun Microsystems, and operating systems such as OS/2, Windows (9x & NT), Unix, OS/400, and OS/390. When DataJoiner is used in conjunction with Visual Warehouse, non-IBM databases, such as those from Oracle, Sybase, and Informix serve as the warehouse database.

5.8.2 Case Study 2: McDonald – Market Capitalization

Mc Donald is a huge organization, working long hours to make sure customers get the quality and service that has become their hallmark. In Canada, more than 1,000 McDonald's restaurants do brisk business. Nevertheless, growing competitive pressures and customers' demand for new values have prompted McDonald's Canada to aggressively expand its market presence with a larger number of strategically located restaurants. At the same time, the company constantly strives to curtail its cost of operations. Extensive discussions among their business executives revealed that they needed detailed, accurate, and timely information for strategic planning and decision-making.

Today, a new DB2-based data warehouse, created and run by IBM Visual Warehouse, is providing key transaction information to market analysts at McDonald's Canada. It includes information that is helping them answer questions such as what combination of products sells the most at a given time of day, which day of the week a new campaign should be launched, the success of promotional campaigns linked with other leading brands, and much more. They were able to achieve enormous returns from their investment in Visual Warehouse. It gave their executives access to information that will help them to substantially increase restaurants sales and reduce operating expenses.

Simplified Development & Reduced Maintenance Costs

The data warehouse, in DB2 for AIX, resides on a four-node RS/6000 SP server. Driving this is Visual Warehouse. Running on a Windows NT server, Visual Warehouse captures transactional data that is scattered on a number of different enterprise business systems, and loads the data warehouse. Says Serge The beauty of Visual Warehouse is that it automates data collection from all the operational systems, refines it and feeds it to the data warehouse. There is no comparable product in the market that combines all these functions in one single package.

Extensive database administration resources are required to manage a complex warehousing system. System Administration is one of the most challenging and expensive aspects of data warehousing. This was one of the primary reasons that the company opted for the product Visual Warehouse. Visual Warehouse takes over the headache of administration and maintenance by providing automated monitoring, performance logging, and other administrative tasks. The cost/performance ratio is also significantly reduced since fewer personnel resources need to be dedicated for warehouse management.

Summary data from the warehouse is extracted onto four department-specific data marts, which reside in DB2 for AIX on different nodes of the RS/6000 SP. Users access the data marts using Cognos PowerPlay and Impromptu clients.

Visual Warehouse supports seamless integration with a wide range of sophisticated decision support tools.

Data Mining

As the data warehouse continues to evolve it will cater to the purchase department's needs, with the restaurant, and operations and franchise departments joining in over the next two years. Information on inventory turnaround, geographical distribution of sales, and restaurant profitability, that will be stored in the data warehouse, will help users from these departments in managing supply chains and planning new locations. The implementation has bought immense benefits to the existing business users. Further the tremendous advantage in mining the newly found data to discover hidden correlation has prompted the organization into reviewing IBM Intelligent Miner.

Implementation Strategy

During the initial planning and testing, data from a combination of 10 restaurants was stored in the data warehouse. This was followed by the integration of a further 100 restaurant and eventually McDonald's aims to get all 1,000 restaurants in the loop. In the meantime, McDonald's will move its data warehouse to DB2 Universal Database Extended Enterprise Edition, as well as migrate to Version 3.1 of Visual Warehouse. Says Edwards, The scalable, parallel-processing capabilities of DB2 Universal Database will help make the best use of the company’s existing RS/6000 system. This combination along with the Visual Warehouse will provide the company with the most scalable and effective system."

The Visual Warehouse Version 3.1 includes new and improved features that make it a powerful data-warehousing tool. The new version makes it easier to make changes to source data and have these changes flow through the business views. Visual Warehouse makes it much easier to change objects and allows changes to be made on the fly. This relieves the database administrator from constantly have to worry about the status of database data. Multiple operations can be performed simultaneously, rather than sequentially. Indexes are

generated automatically, and Visual Warehouse can perform star joins and even display them graphically. IBM was chosen over other vendors because of their technical and support staff were superior to anyone else.

5.8.3 Case Study 3: DAMAN - Extracting Knowledge from Information

As business becomes increasingly information-driven, more and more corporations are moving towards new information system architectures such as data warehouses and enterprise-wide applications. One of the toughest and most frequently under-estimated challenges in implementing these new applications is the timely and accurate conversion and migration of data between legacy systems and the new ones that will replace or augment them. DAMAN Consulting, an IBM Business Partner, is making that challenge much less painful by integrating IBM's business intelligence suite of products into its data migration and conversion solutions.

DAMAN Consulting, a systems integrator based in Austin, Texas, specializes in providing data migration, data warehousing, and decision support solutions developed around IBM business intelligence products. Their approach to data migration and warehousing combines IBM business intelligence products with in-house methodologies. This includes identification of leading edge tools to provide these comprehensive solutions, including IBM DataPropagator and ETI*EXTRACT for data migration, IBM Visual Warehouse for building data marts and warehouses, and Visual Warehouse's Information Catalog (formerly called DataGuide) for meta data management." ETI*EXTRACT, a suite of tools that provides a powerful infrastructure for generating complete extract, transformation, and populate programs, has been developed by Evolutionary Technologies International (ETI), a leading provider of software products for data integration management.

In the data management consulting industry, DAMAN Consulting has earned a reputation for quick delivery of solutions based on IBM's business intelligence products. DAMAN's knowledge and ability to quickly and accurately perform data

migration implementation and IBM's comprehensive package of business intelligence products and integration with complementary products such as ETI*EXTRACT. "IBM's support for products such as ETI*EXTRACT gives them a considerable advantage, because of the integration of the various tools into a single package. IBM's package includes reporting tools, meta data tools, tools for propagating and migrating the data from the source application to the warehouse, and also replication technologies. Having an integrated suite of tools that supports all of the common reporting needs definitely increases the ability to sell such solutions to our customers.

Integrated Products

For DAMAN's customers, quick implementation and conversion from legacy systems to their new data warehouses is a major concern. Using IBM's business intelligence products in conjunction with propagation tools such as ETI*EXTRACT enhances their ability to deliver interface and programmatic conversion solutions under tight deadlines. ETI*EXTRACT's tight integration with DB2 and Visual Warehouse hastens the migration effort and provides the customer an environment that can support their reporting needs very quickly.

One of DAMAN's most recent implementations was for a large property and casualty insurance provider, where the business objectives were to migrate a legacy policy management system to a newer system running on an IBM mainframe using DB2 for MVS. DAMAN also developed several programmatic data interfaces from the new policy management system on DB2 to hundreds of disparate subsystems that support numerous decision support functions. At another site, DAMAN designed a datamart on DB2 for OS/390 to maintain an insurance fraud decision support system. ETI*EXTRACT was used for the data extraction and load processes, and Visual Warehouse was used to design and build the datamart.

"The cost of implementation and deployment is significantly reduced by using the Visual Warehouse prepackaged solution, largely because implementation time is reduced, as is the level of skill required for implementation. The combination of

Visual Warehouse, ETI*EXTRACT, and DB2 delivers a synergy that provides the basis for an advanced decision support environment that is easy to learn, use, and maintain."

Data Migration's Triple Threat: IBM, ETI, and DAMAN.

In addition to providing comprehensive IBM data migration and warehousing solutions, DAMAN has developed tools that assist in meta data management for these solutions, including DAMAN's InfoManager. Meta data is information about the enterprise data and is a critical element in effective data management. DAMAN uses Visual Warehouse's Information Catalog to index the meta data in the application. InfoManager then gathers information from the Information Catalog and other sources of meta data (including disparate applications and data sources external to the organization), and creates an integrated meta data model that can be used for performing impact analyses on the entire computing or decision support environment. The accuracy and currency of the meta data model is further enhanced by InfoManager's Intelligent Agent technology (supporting management of operational activities and change notification).

A big advantage of using IBM's business intelligence solutions is the integration of the meta data arena. Most products are sold independent of each other and then it's left to the clients to create their own integrated model. With IBM's business intelligence suite of products, there's DataPropagator and ETI*EXTRACT sharing meta data with the Information Catalog, which allows users to automate meta data integration. This provides huge benefits in terms of the ongoing maintenance of the data warehouse. In addition to being an end-to-end provider of business intelligence solutions comprised of world class technology, IBM adds value to its partnerships, and that's something that can't be said about the competition.

5.8.4 Case Study 4: Blue Cross & Blue Shield – Real Time Cost Analysis

Blue Cross & Blue Shield prescribes formulas for Real-time Cost Analysis with DB2 OLAP Server and Visual Warehouse. Founded in 1939, Blue Cross & Blue

Shield of Rhode Island (BCBS) provides health coverage for one out of every two Rhode Islanders. This translates into well over 450,000 insurance policies. The requirement of tracking and analyzing every administrative cost associated with selling, implementing, and processing each one of these claims was quite a tedious task. Yet, the BCBS cost accounting department does it every day. Ensuring that all costs are properly allocated across the organization means monitoring business results and interacting with data on a day-to-day, hour-to-hour, and even minute-to-minute basis.

Keeping pace with these demands and the hundreds of thousands of policies was nearly impossible in the past, when analysts had to manually sift through mountains of printed reports and key in the required data into spreadsheets. Today, BCBS has empowered its business analysts with online analytical processing (OLAP) tools to enable sophisticated, multi-dimensional analysis of large volumes of data. Having such capabilities is imperative to BCBS as it allows them to transform data into useful information for analysis and decision-making, and to address the corporate need to understand what is driving business performance.

Since 1993, BCBS, Rhode Island's largest health insurance provider, has been using Hyperion Software's Essbase OLAP server on Windows NT in its cost accounting department. This department plays a critical role in coding, tracking, reviewing, and analyzing all administrative expenses for its 75 diverse lines of health insurance products. Prior to implementing Hyperion Essbase, BCBS cost accountants had to thumb through hundreds of pages of reports to dig up the specific information needed to prepare an expense report and re-enter the data into Excel spreadsheets. Once Essbase was deployed, cost accountants were able to acquire the information they needed almost instantaneously, leaving them more time for their main task--cost analysis.

Recently, BCBS took its OLAP capabilities to a new level by giving its OLAP users direct access to the company's relational data stores with DB2 OLAP Server. DB2 OLAP Server integrates the Hyperion Essbase OLAP engine and

application program interfaces (APIs) with DB2 Universal Database. With IBM's DB2 OLAP Server leveraging the Essbase OLAP engine a great burden was taken off of their IT resources and also helped to give the business user access to vast amounts of summary information that would have normally taken a very long time to produce.

BCBS uses IBM Visual Warehouse to build and maintain portions of the DB2 data warehouse on its S/390 server. There is a large volume of data that needs to interact with other SQL-based decision support tools, and needs to be maintained on DB2. DB2 OLAP Server enables the Essbase OLAP engine to operate on top of this relational store. Accessing the relational warehouse has not come at the expense of its Essbase multidimensional stores. DB2 OLAP Server and the native Essbase system work side-by-side at BCBS, fulfilling important and complementary roles. With DB2 OLAP Server the company is able to do everything that used to be done with Hyperion Essbase, and in addition, have the ability to easily access data from their DB2 data sources.

Blue Cross & Blue Shield uses both DB2 OLAP Server and Essbase as data marts. "Combining Visual Warehouse with DB2 OLAP Server and Essbase provides a managed OLAP environment, in which the process of extracting data from different sources, loading it into the data warehouse, and performing ongoing maintenance of the warehouse could be automated, thereby reducing the need for the IT staff to constantly manage the whole system.

Microsoft Excel is the primary desktop tool for BCBS's business analysts. Predefined Excel templates created internally are used to retrieve the data from Hyperion Essbase or DB2 OLAP Server and automatically load it into the appropriate cost accounting reports that has brought the preparation time by ninety percent. With Hyperion Essbase and DB2 OLAP Server the allocation of costs across the organization were better understood. The effect of the allocations on the final cost objectives, and the impact of administrative expenses on the business were also highlighted. Since the cost accountants now have more time to produce analytical reports and build scenarios, they could give

better feedback and inputs on controlling costs and identifying issues in that process.

5.8.5 Case Study 5: Microsoft Corp. – High ROI

Platform

The Trillium Software System® operates on IBM Mainframes, UNIX systems, AS400 and Windows NT/98. Trillium also runs under Java and is compatible with SAP. Microsoft uses Trillium in both batch and callable mode with Compaq servers running SQL 7.0.

Background

Since its inception in 1975, Microsoft has been the leader in creating software for personal computing. The company offers a wide range of products and services for business and personal use, including operating systems for PC's, server applications for client/server environments, business and consumer productivity applications, and interactive media programs, and Internet platform and development tools. Microsoft products are available in more than 30 languages and sold in more than 50 countries.

Microsoft's sales database contains all sales and revenue data for every product the company sells worldwide. The information is used for everything from general ledger revenue postings to sales forecasts, customer rebates, sales force compensation, and purchase orders for inventory restocking. They need detailed, accurate information on what Microsoft is selling and what Microsoft partners are selling through the channel, down to the individual customer level. Each week, there were 40,000 to 50,000 unrecognized organizations--distributors, resellers, customers. For example, information about a reseller might be entered by more than one distributor with slight variances in the data. Entries may contain erroneous or incomplete data. Revenue totals were not matching up properly for resellers. Report generation was slow and cumbersome, as un-matched records had to be manually reviewed and matched. The rapid growth of the database--14 million new transaction entries each quarter--was outstripping the ability to

correct errors. While records for larger resellers and customers could be manually matched, the proliferation of smaller organizations meant many records were being left unresolved. A third-party data cleansing and reengineering technology was needed. The solution needed to be multi-platform, multitasking, and provide international coverage in multiple languages.

Product Functionality

Trillium Software System is fully integrated into the sales database to clean all new customer data as it enters. They rely and trust Trillium to identify and match to their existing customer list as well as identifying new customers. The software uses modifiable match routines to reconcile transaction records with the appropriate organization. This is done through the use of an industrial strength-parsing engine that utilizes intelligent pattern recognition to identify words and phrases as well as names and addresses. They are then able to review the suggested matches and reconcile accordingly. New customer records are then generated for all transactions that do not meet our business rules for the definition of an existing customer.

Strengths

The software is very tunable, allowing easy configuration of the business rules. It is a worldwide software solution that can process data in several languages. Its scalability is important as the organization expands it to cover other enterprise applications and also move it to smaller systems to handle specific tasks, such as managing customer profiles.

Weaknesses

The current user interface is workable but needs significant improvement. In batch mode, the application is currently based on flat files and is unable to dynamically call a database. This limits the flexibility of integration with disparate products. Lastly, the current application doesn't support double byte character languages.

Selection Criteria

Trillium Software System was selected because of its ability to handle data in eight languages, its multitasking capabilities, and the vendor's ability to provide test files to evaluate capabilities prior to implementation.

Deliverables

Within the first year of implementation the organization experienced a tenfold ROI from their initial investment with Trillium. Revenue and sales reports could be generated in less than two days, instead of five. The number of personnel dedicated to reviewing and auditing data was reduced from eighteen to eight. The IS department was now capable of providing a value-added corporate-wide service by ensuring quality data for channel analysis, revenue forecasts, inventory restocking, general ledger postings and other functions. Because of the over-whelming success experienced with Trillium the organization plans to incorporate the software into on-line applications within their existing departments.

Vendor Support

Trillium provided test files for use in evaluating the system. The test files were detailed and provided assurance that the Trillium system could be adapted to the organizational needs.

5.8.6 Case Study 6: Trans Union – Creation of New Information Based Products

The Company

PerformanceData, a division of Trans Union Corporation, is a leading information and service provider for direct response marketers. With a national name and address list of over 160 million consumers, PerformanceData's file is among the most comprehensive in the business.

PerformanceData understands that organizations need a broad range of information about customers in order to develop a one-to one marketing

relationship with prospects. Direct response marketers from a variety of professions rely on the accuracy, quantity, and effectiveness of PerformanceData’s products and services. The competition to be the best supplier of consumer information is very intense. This competition drives marketing data providers to continually develop consumer lists faster, with better, deeper, more accurate data. PerformanceData recently made a push to reengineer its existing data to find ways of creating new, salient products for its customers.

The Challenge

Rapidly processing large quantities of information is a key to PerformanceData's success. But that information must also be in a format, which is accessible. Initially their inputs were a jumbled mess of data, which their staff could not comprehend. The database team knew it had a wealth of information buried within the comment fields of many of its records. More specifically, they were looking for information on consumers' ownership of large, consumer durables. Most important were boats, recreational vehicles, motor homes and motor vehicles. The challenge therefore was to investigate the comment field, investigate and parse out the valued data components and then create and populate new fields with derived data. The targeted quantity of data was large, but not enormous by PerformanceData standards. The company had 27 million individual records containing the type of information wanted. Fast turnaround of the data (since new data is always in demand) on a Windows NT platform was the goal. They did not want to get into a protracted code writing exercise that would be a drain on both time and resources. PerformanceData needed a versatile data-cleansing solution that could be applied very quickly.

The database team realized several things: they required a solution that could scan free-form text, standardize and transform the extracted data, and create new fields that would be populated with intelligence gathered during the cleansing process. The solution needed to be robust enough to handle large volumes of data and simple enough for expertise to be quickly acquired and

maintained within PerformanceData. This was an important step in maintaining a competitive advantage.

PerformanceData wanted the ability to develop a standardized set of enterprise business rules for data quality management that could be shared across existing and future platforms. Specifically, the company needed a versatile tool that could reach deep within the complex product data and provide repeatable and reusable business rules for data reengineering.

The Solution

The Trillium Software System® was chosen for its ability to rapidly clean and standardize large volumes of generalized data from multiple sources. Trillium's user-defined transformation, data filling capabilities and data element repair facilities are unique among solutions that operate in multi-platform environments. It was Trillium's specific ability to understand, elementize and create a distribution of words and phrases from within floating, free form text that made it a perfect fit for PerformanceData. It was understood that Trillium would not only meet the immediate needs for data reengineering, but that it was positioned for an expanded role in reengineering data in the future.

PerformanceData initially put a team of three into action on the data project. A project manager, a programmer, and a research analyst were selected. The team's first step was to compile a comprehensive listing of boats, RVs, motor homes and all vehicles made and sold in the past ten years, and enter them into tables and parameters for data comparisons.

The Results

As a result of this initial data reengineering project, PerformanceData created a new suite of products for their customers which allowed them to identify owners of specific types of major consumer durables, including boats, recreational vehicles, motor homes, and automobiles. PerformanceData went live with their data-cleansing project within one week of initial training on the system. They were able to identify 14 million records, which had vehicle category information

that could be appended to their customer database. In addition PerformanceData was able to identify vehicles, by type, recode and classify each vehicle in a standardized format, and create and append new vehicle classifications to the original records. Training and learning Trillium occurred easily. They were able to set up the tables and learn how to tune them within one week, and that allowed them to go live immediately.

Because the implementation was so quick and easy, PerformanceData is now considering adding more categories to the mix of products being mined from its consumer database. The company demonstrated that clean data provides a more accurate view of consumers and provides a more valuable product for its clients.

5.8.7 Case Study 7: Cincinnati Financial

The Company

Cincinnati Financial is a holding company whose best-known unit, Cincinnati Insurance Co., markets a broad range of business and personal policies through a 27-state independent agent network. According to Fortune, the company was one of the top 25 publicly traded property and casualty insurers or reinsures in the country--based on revenues. Forbes listed Cincinnati Financial Corporation as the most productive publicly traded U.S. Property and Casualty Company, excluding reinsures.

The company prides itself on its strong customer focus and chooses its agency partners carefully, then focuses on them as customers. At a time when much of the insurance industry is experimenting with direct sales, Cincinnati Financial stands by its agents and believes that the personal touch, which they offer, is a critical marketing advantage. Their reputation for good service and profitability also provides them with a distinct competitive advantage.

The Challenge

High quality service is the key to Cincinnati Financial's strategy for success. It is the key to the company's ability--both to maintain steady prices and maintain high

account retention rates. In the insurance business, a prime measure of strategy is the company's claims service. Although the company already had an excellent reputation here, with increasingly strong competition in the industry, it felt that it could not afford to rest on its laurels. To the customer, the level of claims service is usually measured by the speed in which claims are settled. For that to happen, many pieces in a complex process, involving multiple organizations, must fall into place. For instance, besides claims adjusters, other people and organizations, ranging from legal firms, medical providers, automotive parts, glass companies, and others are often involved in the settlement of just a single claim. With the large number of players involved, there are multiple opportunities for inaccurate information to creep into an insurer's claims database--errors that could easily hold up valid claims.

The opportunity to cleanse the data in its claims master file arose when the company chose to install a new OS/2-based claims application from Policy Management Systems Corp.

The Solution

Cincinnati Financial required a platform-independent cleansing solution that would work in both batch and on-line modes. For instance, the claims database typically received batch feeds daily regarding new policyholders and the names of product or service provider organizations such as loss payees or mortgagees. The primary goal was to cleanse batch feeds as they happened, not after the fact. Another objective was to eliminate database duplication, thereby improving system performance.

Additionally, since the new claims application was designed for client/server, where data was retrieved from a mainframe DB2 database and manipulated by the application on a desktop DB2/2 database, the cleansing tool also had to operate invisibly, in on-line, interactive mode. In normal data entry mode, the claims analyst would enter a name. That in turn generated a pop-up list of policyholders with similar names; the analysts would then point and click on the correct policyholder, and enter claims data.

The company required a tool that was easy to install, maintain, and simple to use. The company looked at several tools and quickly concluded that Trillium was by far the most practical. Trillium was much easier to use, and there was less mystery about how it worked. Trillium's straightforward library of functions would make it easier to customize and maintain in the long run.

The Results

Trillium was implemented by a two-person team, including a senior systems analyst and a senior programmer/analyst, over the course of five months. Additionally, two clerical staff entered updates to Trillium rules tables on a part-time basis during the same period. The entire process was conducted in parallel with installation of the claims application.

Trillium was easily the fastest part of the claims project, and that included the large amount of software testing which the team conducted. Cincinnati Financial went live with Trillium in August 1996. They began by converting policy client tables for the personal insurance lines, business, and claims provider tables. Then, one area at a time, claims are being converted. The success of the headquarters claims project is starting to generate spin-offs. Currently, the company is writing a claims reporting application for adjusters in the field that will send data to the mainframe database. Part of the upload processing will involve cleansing data using Trillium--just another step in Cincinnati Financial's mission to maintain its solid reputation for service.

Summary

Data quality tools are available to enhance the quality of the data at several stages in the process of developing a data warehouse. Cleansing tools can be useful in automating many of the activities that are involved in cleansing the data-parsing, standardizing, correction, matching, transformation and householding. Many of the tools specialize in auditing the data, detecting patterns in the data, and comparing the data to business rules. Data extraction and loading tools are available to translate the data from one platform to another, and populate the data warehouse. In the initial stages of data warehouse development the sources of the data should be examined. Questions should be asked of the data source that would enable the developer of the warehouse to know what problems exist with the data. Once these

problems have been isolated, the warehouse builder could determine which features of the data quality tools address the specific needs of the data sources to be used. The matrix that has been developed will guide the warehouse developer towards the tool that would be appropriate for the data sources that will eventually populate the warehouse. Once the proper tools have been identified, the second matrix could be used compare price, platform, and special features of each tool. The two matrices work together to enable the data warehouse developer to efficiently choose the software tool suitable to the data sources that are to be used in the warehouse.

Chapter Objectives

The objectives of this chapter are:

- To illustrate the importance of Datawarehousing within the Knowledge Management Framework
- To present tools that facilitates the creation of Knowledge Warehouses and Knowledge Discovery within an organization.

Chapter 6

Data Warehousing & Knowledge Management

6.1 Introduction

Knowledge Management involves two kinds of "knowledge": tacit and explicit. A proper data warehousing strategy can play a significant role in the conversion of one form of knowledge into the other to achieve a more productive and usable form.

Tacit knowledge is the knowledge embedded within an individual and includes their ideas, judgments, experiences and insights. The challenge is to articulate what's deeply embedded, and get it out into a usable form" so others can use it profitably. Explicit Knowledge is the articulated knowledge that one can see, read and use. Knowledge management involves the conversion of tacit knowledge and making it explicit.

It is important to realize that Data Warehousing is not Knowledge Management. However there is a lot of implied, or buried knowledge within a data warehouse. This necessitates the use of using OLAP and Data Mining along with Decision Support Tools to find the knowledge that is required. Implicit in all these bits and bytes is the meaning that is not known, but needs to be discovered. Data warehousing plays an interesting role in the knowledge creation cycle. It does not change tacit knowledge into explicit but facilitates the creation of new tacit knowledge by using explicit knowledge.

The contribution of Data Warehouse in a Knowledge Management System is to facilitate the discovery of embedded meaning within the data or information stored within it. A groupware system on the other hand would seek to improve the sharing of tacit knowledge through collaboration and knowledge creation. The knowledge creation is a combination of the above-mentioned two processes. It involves the grouping of people with tacit knowledge and articulating their

knowledge so that it can be used to enrich the existing contents within the databases and the data warehouses within an Organization. This in turn can lead to the creation of new tacit knowledge, which can be made explicit, and repeats in a self-sustaining cycle. The role of data warehouses is to hold explicit knowledge, which helps people create new tacit knowledge and form the basis of real knowledge creation that forms a never-ending full cycle.

6.2 The Data Warehouse Component of Knowledge Management

A through understanding of KM is required to realize the need, importance and the role of a Data Warehouse within a KM system. Useful KM should result in measurably improved business performance by increasing the knowledge of the people in the organization as well as those whom it deals with. The intent is to increase both people's knowledge level and their ability to share it among themselves. These are the two key issues that a KM strategy should address.

Training and other electronic ways such as intranet-based training can address the first issue. The second issue of connecting knowledgeable people together so that they can access the results of each other or share knowledge amongst themselves can be addressed through the deployment of technologies like groupware.

The starting point lies in identifying who is knowledgeable and where their knowledge is stored. This can be done through the use of pointers to people or through documents articulating their knowledge or other repositories within the organization. The HR skills or competencies database or a well-designed search engine can also be used for the purpose. There would be multiple data sources within an organization. These sources have to be leveraged by the organization and used to manufacture better products or deliver better services. These sources of knowledge need to be categorized, classified and stored in well-designed databases to facilitate its sharing and reuse. The real breakthrough in knowledge management is this ability to share. Another way to discover or

uncover knowledge is to search that which is already embedded in files, documents or electronic media, which leads to data warehouses.

6.3 Modeling a Data Warehouse for Knowledge Management [1]

6.3.1 Introduction

Over the years, the information needed to make business decisions has been contained within the glass-enclosed, water-cooled rooms that housed gigantic mainframe computers. The only way to access this information was through the use of dumb terminals and batch reports. These delivery mechanisms were pre-designed and very inflexible. The earliest data warehouse applications, in concept, can be traced back to the use of calculators for producing numbers on the reports from which managers made business decisions. As personal computers became more prevalent, business users started taking any information available and loading it into their personal applications. This time-consuming manual effort often involved reentering data into spreadsheets so that the desired information could be manipulated as needed.

The need for change was finally recognized and became the driving influence behind data warehousing efforts. Instead of requiring the end-user to select and load the information manually, an automated process was developed to extract information and load it into files that were "outside the glass room." Through this data delivery process, data was selected and used to populate target structures commonly known as data warehouses, or more often, data marts (business unit-specific data warehouses). This data was timely, accessible and enabled the end-user to make informed business decisions.

Because of the pressure to deliver these data structures quickly, shortcuts in the development process were often taken. A data warehouse/data mart is supposed to be designed to accommodate multiple queries using large quantities of historical summarized data. In reality, most of these structures are merely images of the production files that were extracted and replicated, i.e. "snap shots" of production

files at a point in time. This extraction reduces the strain of reporting against production files, thus reducing performance degradation, but does nothing to facilitate true adhoc queries. This issue is addressed briefly in the next chapter, which deals with the Microsoft Data Warehousing Framework.

Another result of the pressure to quickly satisfy pressing information needs was that these new structures were populated with whatever data was readily known and accessible to the populator. Instead of taking the time to analyze the proper source of each data item that would be made available in the new data warehouse, any accessible, known source might be selected. An extreme example would be a case in which a number of production reports are selected (as the source), and entered manually into an Access database by the departmental administrator to produce spreadsheets and graphs with no regard to data compatibility.

As the delivery pressure increases, the quality of the information continues to decrease. The need to extract the required information from multiple production systems often becomes too time-consuming and staging becomes too complicated. The risk of providing incorrect information vs. taking the additional time required to determine the most appropriate source of that data is often not even weighed.

A critical success factor for a data warehouse is the perception of the quality of the data it contains. The quality of this data is directly related to the source of the data used to populate the data warehouse. The ability to select the most appropriate source of data for use in a data warehouse is critical to ensure quality as well as provide complete knowledge and understanding of the information made available through a data warehouse.

6.3.2 Data Quality

The fastest way to undermine end-users' confidence in the newly delivered data warehousing "system" is to present data that results in inconsistent, inaccurate, ambiguous query/report results. This can happen as soon as the data in the data

warehouse does not agree with data shown on existing production reports. What causes even more damage is when managers from various business units attempt to make decisions based upon inconsistent results from their individual data marts.

This inconsistency can arise from a number of factors. One frequent problem is that the data used to populate each data mart originates from a different source. There are many data sources within an organization from which to derive a list of customers, product/service offerings, etc. Each of these sources could result in different totals when the numbers are averaged or added. Moreover, even if each data mart uses the same source to populate the data, unless there is a consistent, accepted definition of the information, confusion could, and often does, arise.

Another result of the increased pressure to deliver data warehousing solutions in a short time frame is that data marts are often developed based on a limited knowledge of existing operational systems. Each business unit, or individual department, populates its individual data marts (often referred to as "independent" data marts) with the information they require. The source of their information is that which is within their sphere of knowledge. These contrasts to the use of dependent data marts populated from a predefined set of information that has been integrated from multiple operational system sources. The development of dependent data marts requires some initial data warehousing analysis at an enterprise level. Even though this enterprise-view effort requires more time in initial development, the time spent on follow-on incremental efforts can be sharply reduced. This is especially true when "conformed" dimensions are understood, modeled and utilized.

6.3.3 Data Warehouse – Development

The design of a Data Warehouse should accommodate the requirements of the KM system. As a part of the KM system design life cycle, the information flows and gaps within an organization would have been identified. The Date Warehouse design should be modeled so as to satisfy these critical requirements. As a part of

the analysis the decision making hierarchy, the key decision making areas and the information/knowledge required to facilitate them would be identified. The data warehouse should be modeled to match these requirements.

Even though the single development effort of a complete enterprise data warehouse has been widely accepted as an impossible task, the incremental development of dependent data marts has proven to be feasible and acceptable. In order to understand how these development problems arise, a brief review of the development effort required for a data warehousing effort might help.

The major tasks in this incremental development process include:

- Development of an understanding of the information requirements
- Design of the target structure
- Design of the population/refreshing steps
- Development of initial queries to deliver the required information

The figure 6.1 below provides a simple illustration of the Data Warehousing process as discussed in the previous section.

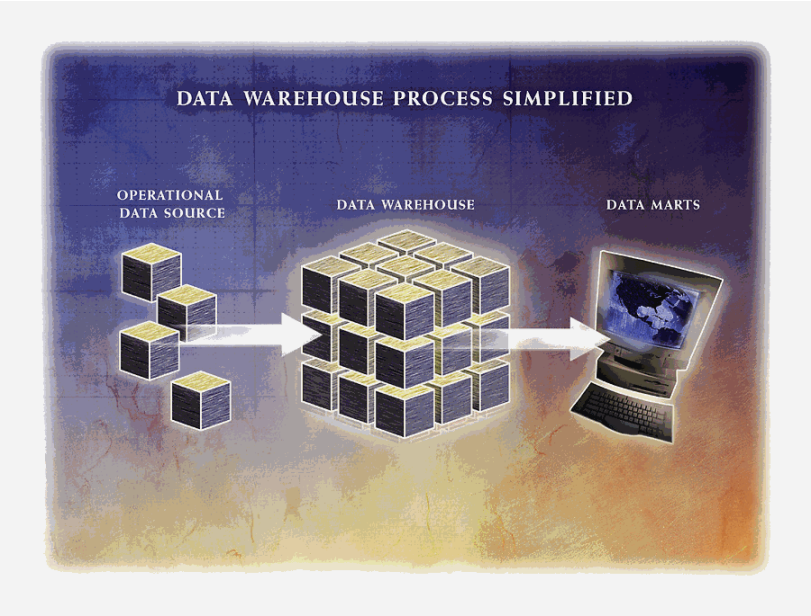


Fig 6.1 Data Warehousing Process

The first step in determining the content of the data mart really consists of two major efforts. The first is to develop an understanding of the meaning of the information requested and to clarify business definitions. It is well understood that as soon as the initial information is delivered, additional requirements will be discovered, hence the idea of incremental data marts. Once the identification of the required information is completed, the source of where that data will come from must be determined. Most often this is accomplished by querying other members of the development team. Using the existing format of the required data, the new target structure for the data mart is designed. More often, these tasks are accomplished using some form of data modeling software. This enables analysis and documentation of requirements and helps design and generates the physical target tables and columns.

Once the information is understood and the new structure is designed, the procedures to populate this data are designed and developed. These procedures include extraction of data from the appropriate operational source, and any transformations, scrubbing, decoding and staging required prior to loading the data into the new target structure. This process can be as simple as extracting the entire production data structure and replicating it, as is, into a new read-only target structure. On the other hand, various sources may be used to populate the new structure, requiring extensive staging procedures to prepare the data. Depending on the incoming quality and format of the data, data transformations and scrubbing may also be performed prior to the final load. This set of procedures is often executed through scheduled procedures developed using an Extract, Transform, Load (ETL) tool. The process of continually refreshing the data in the new data mart can be handled in a number of ways. One is a fairly complex procedure that only adds changes that have occurred since the previous refresh (commonly known as a delta refresh). A simpler method adds all the data that exists at a point in time (a snapshot population). The second method causes the size of the data mart to grow at almost an exponential rate, while the first requires a more in-depth

understanding of why and when data changes occur and how those changes can be captured.

Once the data is available in a new read-only structure, the delivery of this information to the end-user is now available. Through the increased availability and functionality of Business Intelligence (BI) tools, the process of creating queries based on this data has become much easier. In fact, these tools are marketed as capable of being used by non-technical business people, especially to develop adhoc queries. In reality, most organizations create initial queries to satisfy the main requirements of the business end users. These queries can be easily adapted to support parameter-driven requests that will meet the majority of the basic informational needs of the business.

6.3.4 Meta Data

Just as operational applications can become the source for a vast amount of information, data warehouses/data marts can increase the number of applications in the same way. In order to begin to understand the meaning of the information contained in these data warehouse applications, metadata must be available to explain and help facilitate the understanding of the information content. In the past, meta data (whether for operational systems, ERP (Enterprise Resource Planning) applications, or data warehousing efforts) has been thought of as something captured and stored in a repository. Once captured, it can be made available upon request. These capture and distribution activities are usually manual efforts and therefore often overlooked and underutilized.

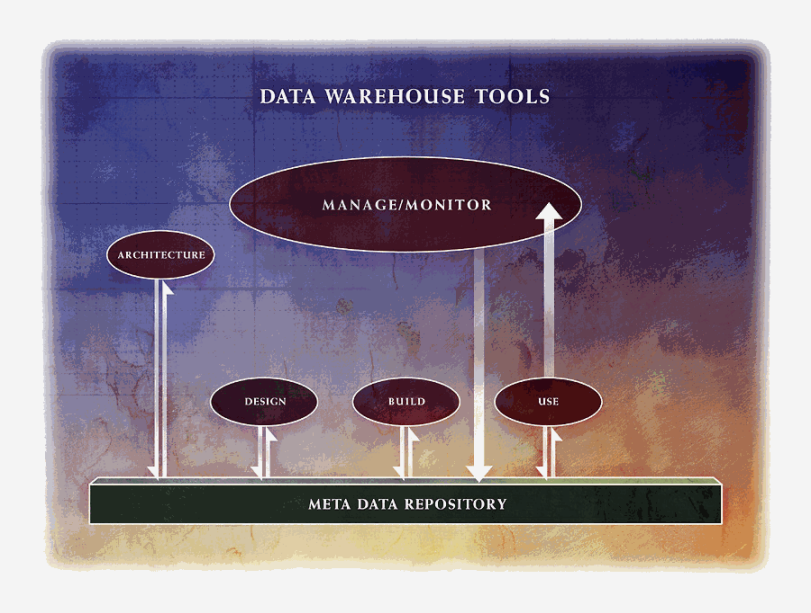


Fig. 6.2 Datawarehousing Tools

Each of the tools used to develop data marts has its own individual metadata requirements. Meta data is not standardized from tool to tool, and meta data was never meant to be shared. In fact, many believe that the subtle differences in this meta data are what give each tool its competitive edge. Individual tool meta data repositories were developed to facilitate the requirements of each individual tool. The fact that the meta data captured in one tool could be used in the next tool (from a tool perspective) is merely a coincidence. Each tool requires some information (meta data) to begin the process as well as to provide information (meta data) when the process it supports is completed. In order to reduce the time required to develop and deliver a data mart, the meta data information must be available and shared between the various data warehousing tools. The above figure 6.2 provides a broad classification of the various tools involved in the Datawarehousing process.

Meta data capture is often overlooked since most meta data is captured through a highly labor-intensive, manual process and its value is not deemed to be sufficient to justify the cost. The ability to electronically capture the software tool metadata available at the end of each logical unit is an absolutely necessity. Additionally,

previously captured metadata should be made available to each tool when work is initiated. As standards such as XML emerge, this integration of metadata requirements between various data warehousing tools should become much simpler.

"The central meta data repository is the heart of the data warehouse. It is used as a "single version of the truth" to provide central definitions of business rules, semantics, data definitions, transformations, and data models. There can only be one version of the truth for these definitions, not multiple, inconsistent definitions…" **[2]**

A number of different potential consumers of metadata exist in a typical organization. These range from the corporate data administrator to the IT staff to the businessperson. Each person has his or her own set of metadata requirements. The capture and distribution of these various categories of information require different procedures and delivery mechanisms. In the past, metadata and the metadata repository were the primary tools of the corporate data administrator. There were volumes of standards and procedures for the capture and usage of metadata within the enterprise. Over time, operational systems metadata has been used more and more by application developers for impact analysis. The repository has made it possible to understand the impact of a change to an operational system. It also has provided a picture of how various applications relate to each other. This use has expanded the scope of the repository and its metadata contents to encompass the IT organization requirements, even though its use was very limited. Most of the metadata contained in the repository to support these needs was about technical artifacts such as applications, jobs, files/tables, records/columns, reports, queries, etc. Little if any business metadata was available (or deemed to be necessary).

As more and more business knowledge is required, the need for business metadata becomes more obvious. The content of the repository begins to be extended to include not only the operational systems technical metadata, but also

the definitions, meanings and business rules (i.e. business meta data) that provide the business knowledge behind these system components. The capture of this information is a bit more difficult. No one has yet figured out how to transfer knowledge that resides in a businessperson's brain into a machine-readable format (brain scan). Often the only electronic version of this knowledge is in a word processing document or spreadsheet, or possibly in a data modeling tool. It has become imperative that this information be captured electronically for retention in the repository. Once the problem of capture is overcome, the delivery vehicle for this new meta data to the business end user has to be established. It is important to realize that the metadata currently provided through most software tools, including most meta data repositories, focuses on technical content. As the distribution of this metadata is expanded to include business metadata and business end-users, the content of that information should be more business-oriented.

An Organization is faced with newer challenges related to e-Business relative to metadata. An organization must be prepared and enabled to provide metadata to users within organizations outside their own. The business end users, as we know them today, are only a small percentage of the audience of the future.

There is one other group of metadata consumers. This is not a group of people, but a set of software tools. Restricting the discussion here to data warehousing, many tools are available to support the development of a data warehouse/data mart. As the use of each tool is introduced and established, the metadata requirements for that tool must be identified. Previously captured metadata should be electronically transferred from the enterprise metadata repository to each individual tool. In return, the tool-specific meta data should be analyzed for inclusion in the enterprise meta data repository. If captured metadata is never required by another tool, or group of users, a question should be posed as to whether or not the storage of this information is really required. As the consumption of metadata grows beyond the organizational boundaries, information requirements also grow.

6.3.5 Dimensions

As the pressure to deliver incremental data marts at high-speed increases, organizations look for ways to reduce development time. One way that has proven successful at not only reducing the development time, but also increasing the quality from one increment to another, is the use of conformed dimensions. A conformed dimension is a dimension that has been analyzed and modeled by a central data warehouse design team and then used for all following data mart implementations. The official definition of a conformed dimension, according to data warehousing expert Ralph Kimball, is "a dimension that means the same thing with every possible fact table to which it can be joined." For example, once a product dimension has been analyzed and modeled, that knowledge can be reused. Because the actual usage may be different, the actual physical structure of this dimension may vary from data mart to data mart, but the logical structure, and imbedded business rules, need only be developed once. As more and more data is requested regarding a dimension, those items can be added to the logical model and are then available for future increments and enhancements. In addition to the physical model of each dimension, the business meta data, including business rules and approved sources, can be maintained in a meta data repository.

Another analysis effort, which needs to be done only once, is to develop an understanding of the effect of business events on the changing of information in each dimension. This is commonly referred to as "slowly changing dimensions." Rather than simply replacing the entire product dimension, for example, on each refresh cycle, the business event that adds, changes, or complements a product is analyzed. A determination is then made as to how best to reflect the change in the dimension. When this business event occurs, the resulting information could be captured and used to refresh the dimension. Using this approach, the amount of data that is captured and loaded on each cycle is reduced.

6.3.6 Data Warehouse Sources

The underlying requirement that satisfies both data quality and conformed dimensions lies in the definition of the appropriate source for each data item that is included in a data warehouse implementation. There is no questioning the fact that each and every piece of business information is contained in multiple physical files/tables throughout the organization. Some of these physical structures are referred to as "master files" while many are extracts that have been developed and used for specific purposes. In order to facilitate the "correct" choice of data sources, an extensive analysis must be undertaken to:

- Determine which files are to be considered as master files
- Map the business elements to the most appropriate location in the enterprise.

This knowledge should be gathered by either the enterprise data administration organization or the central data warehouse team. Once gathered, easy access to it must be provided to the rest of the organization. The capture, maintenance and distribution of this information should be facilitated by an enterprise metadata repository that is available to all individuals within an enterprise.

The task of developing this "business element-appropriate source" link is monumental. It need not be undertaken as a single effort all at once. In fact, it can be handled concurrently with the identification of the requirements of each incremental data warehousing effort. As the need for individual business elements are identified, the appropriate source can be determined and provided to the data warehouse developer. The identification and capture of not only the actual physical field/column and file/table of the appropriate source, but also the application that maintains this information, is important. This valuable information can then become another set of metadata attributes about a business element. In organizations where modeling tools are used to facilitate the design of each data warehouse/data mart increment, a list of business elements, appropriate source, business rules, business definitions, previously developed data models, etc. can be electronically transferred from the enterprise meta data repository to the modeling tool. The

presence of this type of information saves discovery time at the initiation of each new effort. It also helps ensure data consistency and quality whenever the information is presented.

6.3.7 Enterprise Repositories

The real difference between the metadata repositories contained in each data warehousing tool and a true enterprise metadata repository is the scope of the information contained. The tool repositories are meant to support the functionality of the tool and therefore contain only that information required by the tool itself. Very seldom is additional information captured in these tools (unless through user-defined fields).

On the other hand, an enterprise repository is designed to provide a source for managing the "metadata" for the entire enterprise. As one data warehouse professional remarked, "metadata is not data about data, but rather, data about resources." As a result, many enterprise repositories contain much more than just the typical information about the information assets of an organization (such as applications, files, programs, files,) and have begun the capture of additional technical and business metadata. Many organizations use the Zachman Framework (a framework for information systems architecture) to help identify the types of metadata that could be captured and maintained in an enterprise repository. Today an organization would be lucky to have even the first three columns of the last row populated "to an excruciating level of detail" (which means including textual definitions/descriptions!). As the need for this information increases, the pressure to capture and deliver the entire contents of the Zachman Framework increases. The value of the understanding and reusability of this information is impossible to measure until the need arises and the information is not available.

6.3.8 Conclusion

As organizations continue to push the decision-making process further and further out in an organization, the greater the need for understanding becomes. There are fewer and fewer individuals who can be contacted to answer the question "what does net sales mean on my sales report?" Wouldn't it be nice if, when that question arises, that the business person could right-click on that net sales number and view a box that describes (in the business user's language) the definition, business rules, calculation, application source of the data, time last updated, etc.? And as normal procedure, that every report and query that shows net sales would contain the same value? It need not be impossible or improbable.

6.4 Data Warehouse Design Guidelines [3]

Many data mart vendors have lured organizations into building the data mart before the corporate data warehouse was built. In other cases the data mart vendors sell directly to the finance, marketing or sales department, bypassing any architectural considerations brought up by the IT department. Organizations discover that an architecture made up of data marts with no corporate foundation of a data warehouse leads them to massive waste and dissatisfaction with their DSS environment.

There are several approaches that can be taken. One approach is to ignore the problems of having no corporate data warehouse as a basis for data mart processing. This leads to the generation of redundant data across each data mart. First one data mart then another duplicates the same detailed corporate data. Furthermore, the redundancy comes at the worst place - at the detailed data that each data mart needs to do their own unique processing. Typically the same sales data, order data, and customer data at the detailed level are spread across many data marts. The cost of trying to support massive amounts of redundant detailed data is not something that an organization can ignore,

- There is no integration between the different data marts. The data present in the data marts of individual departments would be inconsistent with the

other departments. This leads to non-uniformity of data across the organization.

- The interfaces required to move the data into the many data marts are many and burdensome. Each legacy application requires its own unique set of interfaces to each data mart. If there are m applications and n data marts, then m x n interface programs are required. The manpower required to create and maintain the individual interfaces from the operational systems into the data marts is staggering. The maintenance required once the interfaces are built is even more staggering.

But the classical problems associated with building data marts first are not the only frustrations the corporation faces. Another disappointment is in the technology used to manage the data mart. The data mart built with no proper foundation of a data warehouse tends to grow very rapidly. Soon the data mart technology that is quite comfortable handling one volume of data is stretched beyond recognition as it struggles to cope with volumes of data and processing for which it was never designed. The build up of these problems grows over time, and it is only a question of how long an organization can hold out before it stumbles beneath the burden of woes presented by data marts with no proper foundation.

There are several alternatives for the organization once the decision is made to build the proper foundation for the data marts. The three practical alternatives are as mentioned below:

1. One option would be to discard the existing structure of inconsistent data marts and begin the process again.

   This approach presents the data warehouse or the DSS architect with the most precious of opportunities - of starting with a blank slate. In this regard, this alternative is a very good choice. And in some cases, this is the only real alternative. Unfortunately this alternative negates the earlier time, efforts and investments made.

An interesting question that relates to this alternative is whether throwing the data mart away and starting all over again is necessarily a bad idea. There is much that is learned in the building of the first iteration of the data mart. That learning experience will hardly be wasted even though the data mart itself may be physically discarded. So starting over from scratch and using the experience gained by building the first version of the data marts may not be such a bad idea after all.

2. The other option would be to design a data warehouse using the existing data marts within the organization.

   This alternative is in trying to save everything in the data mart environment while building the data warehouse. Common sense defies this approach since the reason for building the data warehouse was that much of the data mart was not salvageable. Therefore this alternative doesn't make a lot of sense.

3. The third option would be to retain some portions of the existing data marts, while discarding the others and build a data warehouse.

   This option assumes that at least some of the data mart may be easily salvageable. The reports, the queries, and the screens that appear before the end user in the current data mart environment probably are salvageable, at least to some extent. And some of the basic data - typically summary data - found in the data mart may be useful in another reincarnation of the data warehouse/data mart environment.

   One variation of this approach is to try to use a data mart as a basis for a real data warehouse. The problem is that much of the existing data mart will prove to be very customized for the department that owns the data mart. Because the data mart is peculiar to a department, it is not easy to try to stretch the data mart into a truly corporate configuration. In most cases the fit is very uncomfortable and very unrealistic. There is only so much that can

be salvaged out of a data mart when the decision to build a proper foundation is made.

It is probably not realistic to think about trashing everything in the data mart environment when the data warehouse is built. By the same token, it is not realistic to think that all that much can be salvaged. The best an organization can do is to attempt to salvage as much as they can.

Once the organization discovers that a data warehouse must be built the steps to achieve this must be clearly understood. There is no real difference between building a data warehouse as the first step in DSS and building a data warehouse as an afterthought after the data marts have been built insofar as the proper steps of construction are concerned.

Some of the considerations about building the data warehouse are:

- Using a data model as a road map,
- Building the data warehouse incrementally, in small fast iterations,
- Involvement of the end users
- Being prepared for massive amounts of data
- Recognizing the importance of metadata as an active part of the data warehouse, DSS infrastructure,
- Understanding the expectations of the end users.

The steps in building a data warehouse have been discussed in conferences and articles in great depth in the past few years. There are many articles and books that describe the work that needs to be done during the construction process. There is no shortage of excellent material that can be easily and cheaply obtained. Using one or more of these plentiful and inexpensive sources is the place to start the real data warehouse adventure, once the organization discovers the fallacy and inadequacy of the data mart first approach.

6.5 Data Warehousing and e-Business [4]

Over the last several years e-business and its associated arms e-commerce, e-solutions, e-economy have revolutionized the industry. Most of the organizations are faced with difficulties in adapting their systems to provide e-solutions. These Organizations spend considerable amounts of development to create an e-Business solution that accomplishes nothing much in terms of providing value addition to their existing products, services and customers.

6.5.1 e-Business

Too often organizations believe that e-Business is a cure-all to their selling issues. They perceive e-commerce a nothing but building a website, marketing it using the popular search engines. Unfortunately reality offers a vastly different picture. e-Business is nothing more than a new distribution channel/customer touchpoint (the internet) that serves a new, but familiar customer segment (web surfers). Typically many of these web surfers are not new customers but existing customers that have internet access and are inclined to make purchases through the internet. If these customers have a negative experience on the organizations website then this can directly impact sales in other distribution channels (store, catalog, etc.). As a result the organizations website, if incorrectly built has the capacity to reduce the current customer base.

An Organization needs to examine their business model to make sure that their e-initiative fits the model. After examining the model the organization may decide that to significantly alter it, which would initiate many changes within the organization. The organization would also need to make its presence felt on the internet through the development of a intuitively designed web site. While the website might not generate a single on-line order it can create leads that can differentiate the organization from the competition.

6.5.2 System Integration

One has heard many stories on the various web problems that corporations have encountered and read about companies that could not fulfill all the orders that they

received during the holiday season. There is an old marketing rule that "A customer will tell 2 - 4 people when they are happy with a product/service you've given them. These same customers will tell 12 – 20 other potential customers when they are dissatisfied with a company's product/service. The following example illustrates the problems faced by most corporations/companies/organizations that have entered the booming e-commerce industry. In the quest for gaining competitive advantage by becoming one of the first companies to offer on-line training most company fail to make sure that they have the right infrastructure/resources/expertise to back them. A customer who looked to purchase a jacket from a favorite apparel designer for an occasion in the family figured that it would be easier to visit the apparel designers website. Upon going to this website he has successful in locating the exact coat that he wanted, unfortunately when he tried to place this order on-line he was told that the coat was out-of-stock. Undaunted he then called the apparel designers at their office and was told that they were out-of-stock on this particular jacket. However, a couple of days later he was in his local mall and walked past one of their stores. On a whim he entered this store and discovered a rack full of the exact same coats he saw on the web. However these coats were priced at Rs.500 less than the website.

What happened is that the apparel designer in their rush to build a corporate website did not take the time to integrate their e-Business system into their existing legacy order entry systems. It is important to note that most of the web debacles that are recorded have occurred with the well-know Global 2000 corporations and not with the new "dot com" startups. This has occurred because the established companies have to reengineer their existing systems to work with their e-Solutions. On the other hand, the "dot com" startups did not have any existing systems to be integrated.

6.6 Knowledge Discovery and Warehousing

6.6.1 Introduction

Leveraging intangible assets is one of the most critical business issues of this decade. This necessitates the free flow of ideas, insights and knowledge within an organization along with a high degree of trust and requires a great deal of nurturing and facilitating. The organizations have to leverage knowledge effectively, without which they face the risk of being eliminated from a highly competitive business environment. This involves a radical shift from investing in technology alone and brings about an unparalleled velocity of change and innovation that involves investing heavily in training and education to help organizations survive in the fast-paced, knowledge era. It is imperative that knowledge workers must constantly upgrade their knowledge and skills in order to thrive in the new economy. Learning is not only important but is vital and requires time out from being engaged in productive activity. Some companies have devoted a fixed percentage of payroll or revenue to training and development. This necessitates the need for meaningful metrics of performance and demonstrations of payoff.

The following are the two most important constituents of a Knowledge Management system:

i) **Knowledge Warehousing**

ii) **Knowledge Mining**

The above-mentioned constituents are explained in the following sections using a commercial Knowledge Management System (KMS) Dataset, which is a product of Intercon Systems.

Knowledge Warehouse is a traditional Relational Database; used to manage structured data (fields, properties), expanded also to manage none structured texts (Word processor documents, RTF, Excel workbooks, Text files, messages, etc.). Knowledge Items are compressed to about 25% of original volume, still maintaining their formatting attributes. Depending upon the complexity, scope

and the requirements of an organization the Warehouse would need to be scaled to a multi dimensional structure.
Knowledge Mining extends the familiar Data-Mining concepts into wider scope: analyzing relationships between data and unformatted text, to target, locate, connect, navigate and extract knowledge from text.

Knowledge-Mining tools are used to generate lists of selected Items that match multiple criteria, from among the Items in Store. Knowledge-Mining navigation tools provide details on the listed Items. User may browse Items by pointing the mouse on navigation maps. Preview pan displays Items' text conforming with multiple formatting protocols. New search methods and navigation tools provide powerful navigation methods within text. DynaLink extends the familiar Hyperlink concept into flexible; user defined dynamic linking between Items. Derivative text can be extracted from Previewed text, copied into destination applications such as MS-Excel. Items in Knowledge store can be copied and opened in MS-Word. Text segments may be copied via the clipboard into word. Multiple segments can be 'nailed' on 'spike', then copied into the Word document.
Data Warehousing and Mining applications are mainly run by big organizations that generate large volume of transactions. KMS on the other hand fits a broader range of professionals ranging from:

i). Anyone who needs flexible tools to manage their information

ii). Professionals who need improved control over their document-files (secretaries,

lawyers, reporters, marketing directors, students)

iii). Information professionals (Corporate: personnel, marketing, documentation,

contracts, orders departments; Researchers: clinical, intelligence, academic.

iv). Librarians

v). Magazines: editors, reporters, administrators) with data management
requirements on textual materials;

vi). Corporate users who manage and access thousands to millions textual Items
(financial analysts, R&D departments, public relations, internet information
providers).

6.6.2 DataSet

DataSet saves disk space, organizes the documents, improve back-up procedures and helps an organization to find all the Knowledge that is required. DataSet combines Relational-Database (RDB) paradigm with Focused Information Retrieval paradigm. RDB is used to manage storage Items and their properties within Dataset's Stores. Relational database technology provides tools for processing large amounts of both general and specific information related to particular topic. Dataset's proprietary technology sets new effectiveness standards for RDB data targeting. RDB technology is supplemented with Dataset's unique capabilities to manage text. DataSet provides comprehensive search and retrieval tools, which can locate Items almost instantly, by words, phrases and much more; interrelationships between stored items are identified, providing tools that allow navigation through text, with unprecedented ease and accuracy.

For example: Store containing customer-items stores properties such as names, addresses, phone-numbers, past purchases payment histories, and more. Such information is best handled applying RDB's SQL queries with Dataset's extensions, to locate relevant records. RDB methods also provide ability to cross-tabulate items, acquiring more insight into relationships between data elements. That same Store also contains customer textual documentation - letters, notes, reports etc. within single system and one user interface, facilitating Knowledge access.

With DataSet one can keep the information corpus (text and records) in several independent Stores. Collection of Stores is managed within a common Text-Warehouse. Complex queries can be developed, employing combination of property queries (similar to SQL) and textual knowledge mining techniques. DataSet provides capability to perform complex queries using powerful, yet simple hierarchical query processes:

- Perform search within textual properties
- Perform search on properties using relational operators
- Perform search on full text using Boolean operators AND, OR and NOT
- Perform search applying Fuzzy Technology
- Perform search for words and phrases within document
- Receive query results in and answer-set, ranked by proximity completeness, frequency etc.

DataSet applies Single user, Networked user and Client/Server paradigms. The system is composed of two logical units:

- Store background engine, is the 'data-pump' and item's repository.
- User Interface foreground engine, that perform the computational intensive tasks, such as interactive query building, query evaluation, query results computation and display.

Shifting the computational load from background process to the foreground, allows Store repository to be optimized to service large volume of simple queries, utilizing computational resources at user workstation, where resources are practically unlimited

Following are some of the important features of the Data Set's Search Engine:

- **Interactive Query building process:**

Query composition is incremental; each returns visual feedback, reporting on steps' effect on query results.

- **Powerful Set of Properties:**
  Classifier, associate items within logical groups; Specifiers, provide multiple alternative keys for items' targeting. Collection of such properties is user-defined for each Store and knowledge source.
- **Hierarchical Query and Networked linking procedures:**
  Classifiers may be applied on top, to identify and targeting groups of documents; Specifiers, are applied on classified items activating fuzzy-search methods. Full-text Search of classified and specified targeted documents is done. Navigation between items is facilitated by DynaLinks and AntiLinks.
- **Graphical User Interface:**
  DataSet visualizes intermediate results with DynaGraph, an interactive graphics tool that summarizes knowledge graphically and prompts for informed action.
- **Interactive, Full-text Search query composition:**
  Fuzzy selection of criteria-words, from the list of words actually in store; expanding query scope with analogues (synonyms antonyms etc.) and derivatives (word stems etc.). Feedback on each criteria-word, assessing its 'query quotient' (effectiveness).
- **Multidimensional graphical query evaluation:**
  Full-text query results are represented graphically, evaluating frequency distribution between queried words.
- **Compression:**
  Stored items are highly compressed, saving up to 80% of disk space and communications processes.
- **DynaText Preview:**
  Universal text-Preview, displays within DataSet of native WORD, EXCEL, RTF, HTML and TXT files. DataSet integrates information retrieval procedures, Items' storage management and Item's display.

- **Textual Navigation:**
  Navigation within and between documents is facilitated by graphical 'knowledge maps'. Documents are browsed either sequentially, or by following mapped knowledge-value factors, or by sections' targeting or DynaLinks.
- **DynaLink and Find:**
  DataSet's unique 'Find' functions allow navigation within items with multiple simultaneous treads.
- **MS-Office integration:**
  DataSet is tightly integrated with MS-Office applications. DataSet extends OPEN and SAVE menu functions into broader, significant paradigm. DataSet as 'save' agent for MS-Office, adds meaning and accessibility to stored contents. 'Open' services range from simple service of document to comprehensive compilation of knowledge, availed to MS-Office through clipboard or Spike.

6.6.3 Fuzzy Queries

DataSet supports fuzzy queries, searching for words and text similar to those entered in query restriction. Rather than looking for exact matches, DataSet modifies query words, optionally looking for modified forms. Fuzzy query extends the query in the following ways:

- Wildcard matching of queried word, setting 'fuzzy-level' control interactively.
- Use of Analogues (Synonyms, antonyms etc.) to broaden scope for each criteria word.
- User's generated variations of words to broaden queries (e.g. "sink" can be automatically expanded to include "sunk", "sunken")
- Fuzzy expression matching of Classify and Specify criteria.
- Dynamic normalizing of categories of within classifier properties.

- Weighting factors can be dynamically set. Weights are used to rank query results.

Dictionary is a file, with DCT extension, used by DataSet to list words in stored documents. Locator is a file, with LCT extension, that relates (points) between words in DCT file and Items in store. Index is a file with STM extension, storing data structure used for fast, Fuzzy access to documents with specified properties - another unique and powerful DataSet's extension to traditional SQL technology.

6.6.4 Indexing & Search

DataSet's indexing and search by properties with classifiers and Specifiers are independent of the full-text search. It is possible to structure query incrementally, beginning either with Classifiers, Specifiers or full text, processing the result-set with additional search method. Corpus refers to the entire set of documents available for inclusion in your system. Scope, refers to a set of documents searched during a query. Scope is limited to documents stored in a single Store.

6.6.5 Scanning, Filtering and Indexing

Scanning, filtering and indexing are closely related steps in the process of building the structure used by DataSet to satisfy query requests. Scanning is the process by which DataSet identifies files within the files stored on the disk (F3 Tab-Scanning). Filtering is the TAB used to extract knowledge from document into properties. Indexing is the process by which knowledge, extracted from documents is stored.

6.6.6 Sentence Breakers and Paragraph Breakers

Sentence Breakers are characters that are used to determine end of sentence in a document. Sentences are the elementary structure of text where text criteria are evaluated for proximity. DataSet converts unstructured text files into RTF format, analyzing sentence breaks, paragraph breaks and section breaks.

Breaking such elements accurately contributes to accuracy of 'relevancy' assessment for an item.

6.6.7 Result Sets

Specific, incremental limiting conditions are used to limit the number or returned items and the type of returns.

Result sets are controlled by:

- Interactive Query process incrementally displays the number of potentially returned items, while query is assembled.
- Table of answer-set, tabulating documents properties, text relevancy score and section valuation, before actual items are selected for preview.
- System adjustable default sets the maximum number of items that are tabulated in response to query.

6.6.8 Storing

While storing unformatted Items, DataSet optionally formats the text. Conversion identifies within the text sentences, paragraphs, indents, numerators, tabulation, and tables. Converted text, with embedded layout instructions is saved in RTF format. Optionally, DataSet identifies quantitative tabular data, and copies the table into structured EXCEL data object.

6.6.9 Networking Environment

DataSet provides solutions in to wide spectrum of situations:

- Single user environment, where Store manager and Query manager, all reside in one machine;
- Networked, Multi-user environment, where multiple users share networked Store. Copy of Query manager is on foreground of every user, Store manger runs on each user's system, Shared Stores reside on network accessible file system.

- Client/Server networked environment, where single background process runs Store manager for data-pumping, while multiple Query managers are in foreground processes on each user's system.

6.6.10 Store Content Updation

Update method is environment dependent:

- In single user environment, Stores are opened exclusively, without restrictions on updates.
- In Networked environment, users open a store either as 'read only', 'exclusive' or 'shared' mode.
- In Client/server mode, Store is updated in background - OLE server process. However, only one document can be stored in the system on any time.

6.6 11 Store Property Types

Multiple properties can be defined. Each property attributes can be set as follows:

- **Title** Any descriptive text, up to 255 characters.
- **Class** Text descriptor, common to multiple Items. i.e. : Manufacturer name, a common

  descriptor for multiple Products. Property contents are stored in alias table.
- **Text** Any descriptive text, up to 2047 characters.
- **Keywords** Sequence of words and expressions that typically define the Item.
- **Options** Set of options depicting the Item.

- **Check box** Set of options depicting Item. Multiple options can be selected
- **NumericID** Identifier such as SSN, Catalog No.)
- **Quantity** Use to store additive data.
- **Telephone No.** Formatted area code, extension identifier and number.
- **Document date** Formatted date.

6.6.12 Store specifications

User specifies properties created and maintained in a Store. User decides when to enter new Items into store. User decides on mode and timing of property capture. Items can be stored directly from File |Store function of text originator application.

Typically, all transient objects are stored with the Item. When Item is opened by its source application, all transient objects are available.

6.6.13 Item Properties

The features hereby were designed for easy property entry:

- Items are displayed in preview pane, below the properties display pane. Contents for property can be selected from previewed text efficiently, copied to correct property field by hot-key entry.
- Property 'header-in-text' can be defined for a property. 'Header-in-text' is automatically searched for, when Item is previewed. Text following the header is automatically formatted and copied to the property field.
- Date, Phone and similar Properties, are located in Item, normalized and copied to the respective property field.
- System properties, such as 'title', 'date', 'characters' etc., are generated automatically.

Summary

There are two key reasons to expect the management of knowledge to become an increasingly important issue in the corporate world over the next few years. One is the imperative to accomplish “more with less” in the wake of downsizing and restructuring of organizations; the other is the need to strengthen relationships with customers.

Empowering employees with more knowledge can help them provide faster and better quality service to their clients. Instead of duplication of efforts a knowledge base of concepts and information helps employees build on previous work and customize solutions according to the client’s needs.

Knowledge strategies require proponents and facilitators in order to succeed and an organization might require the services of consultants who specialize in a particular product/service line, capability, industry or geography. In this capacity, the consultant organizes, monitors and facilitates the flow of knowledge for a specific community of practice or a group that shares a certain type of knowledge or expertise.

Technology is one of the key enablers that facilitate the building and deployment of Knowledge Management and Knowledge enabled systems. Building of efficient and properly designed Databases and Data warehouses form one of the key requirements for the development of Knowledge repositories within an organization. These repositories form the key sources of knowledge and in conjunction with Knowledge Discovery or Online Analytical Processing (OLAP) services help the managers within an organization take timely, effective and quick decisions that bring tangible benefits to the organization and help realize their goal.

References:

[1] Greta Blash - Rochade
The Knowledge Advantage Maximizing Data Quality In The Ata Warehouse Through Enterprise-Wide Understanding

[2] Pieter R. Mimno, The Data Warehouse Institute

[3] The Data Administration Newsletter (Tdan.Com)
Robert S. Seiner - Publisher And Editor
Source URL: http://Www.Tdan.Com/

[4] Ref The E-Dilemma
David Marco - Enterprise Warehousing Solutions, Inc.

References:

[1] Source URL : http://www.techguide.com

References

1. http://www.allinterview.com/showanswers/160457.html
2. http://www.information-management.com/issues/19990401/96-1.html?type=printer_friendly
3. http://infonitive.com/?p=205
4. http://www.ibm.com/developerworks/data/library/techarticle/dm-0507gong/
5.
6. http://www.scribd.com/doc/49205932/Data-Warehouse-Architecture-Components-and-Implementation-Options-Infonitive
7. http://www.donmeyer.com/art1.html