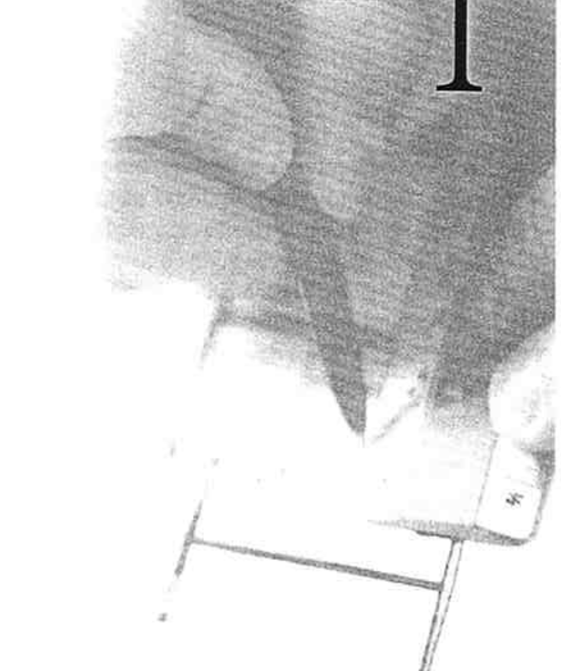
CHAPTER 1

Introduction



Welcome to the first book that thoroughly describes the data modeling techniques used in constructing a multipurpose, stable, and sustainable data warehouse used to support business intelligence (BI). This chapter introduces the data warehouse by describing the objectives of BI and the data warehouse and by explaining how these fit into the overall Corporate Information Factory (CIF) architecture. It discusses the iterative nature of the data warehouse construction and demonstrates the importance of the data warehouse data model and the justification for the type of data model format suggested in this book. We discuss why the format of the model should be based on relational design techniques, illustrating the need to maximize nonredundancy, stability, and maintainability. Another section of the chapter outlines the characteristics of a maintainable data warehouse environment. The chapter ends with a discussion of the impact of this modeling approach on the ultimate delivery of the data marts. This chapter sets up the reader to understand the rationale behind the ensuing chapters, which describe in detail how to create the data warehouse data model.

Overview of Business Intelligence

BI, in the context of the data warehouse, is the ability of an enterprise to study past behaviors and actions in order to understand where the organization has been, determine its current situation, and predict or change what will happen in the future. BI has been maturing for more than 20 years. Let's briefly go over the past decade of this fascinating and innovative history.

You're probably familiar with the technology adoption curve. The first companies to adopt the new technology are called innovators. The next category is known as the early adopters, then there are members of the early majority, members of the late majority, and finally the laggards. The curve is a traditional bell curve, with exponential growth in the beginning and a slowdown in market growth occurring during the late majority period. When new technology is introduced, it is usually hard to get, expensive, and imperfect. Over time, its availability, cost, and features improve to the point where just about anyone can benefit from ownership. Cell phones are a good example of this. Once, only the innovators (doctors and lawyers?) carried them. The phones were big, heavy, and expensive. The service was spotty at best, and you got "dropped" a lot. Now, there are deals where you can obtain a cell phone for about $60, the service providers throw in $25 of airtime, and there are no monthly fees, and service is quite reliable.

Data warehousing is another good example of the adoption curve. In fact, if you haven't started your first data warehouse project, there has never been a better time. Executives today expect, and often get, most of the good, timely information they need to make informed decisions to lead their companies into the next decade. But this wasn't always the case.

Just a decade ago, these same executives sanctioned the development of executive information systems (EIS) to meet their needs. The concept behind EIS initiatives was sound—to provide executives with easily accessible key performance information in a timely manner. However, many of these systems fell short of their objectives, largely because the underlying architecture could not respond fast enough to the enterprise's changing environment. Another significant shortcoming of the early EIS days was the enormous effort required to provide the executives with the data they desired. Data acquisition or the extract, transform, and load (ETL) process is a complex set of activities whose sole purpose is to attain the most accurate and integrated data possible and make it accessible to the enterprise through the data warehouse or operational data store (ODS).

The entire process began as a manually intensive set of activities. Hard-coded "data suckers" were the only means of getting data out of the operational systems for access by business analysts. This is similar to the early days of telephony, when operators on skates had to connect your phone with the one you were calling by racing back and forth and manually plugging in the appropriate cords.

Fortunately, we have come a long way from those days, and the data warehouse industry has developed a plethora of tools and technologies to support the data acquisition process. Now, progress has allowed most of this process to be automated, as it has in today's telephony world. Also, similar to telephony advances, this process remains a difficult, if not temperamental and complicated, one. No two companies will ever have the same data acquisition activities or even the same set of problems. Today, most major corporations with significant data warehousing efforts rely heavily on their ETL tools for design, construction, and maintenance of their BI environments.

Another major change during the last decade is the introduction of tools and modeling techniques that bring the phrase "easy to use" to life. The dimensional modeling concepts developed by Dr. Ralph Kimball and others are largely responsible for the widespread use of multidimensional data marts to support online analytical processing.

In addition to multidimensional analyses, other sophisticated technologies have evolved to support data mining, statistical analysis, and exploration needs. Now mature BI environments require much more than star schemas—flat files, statistical subsets of unbiased data, normalized data structures, in addition to star schemas, are all significant data requirements that must be supported by your data warehouse.

Of course, we shouldn't underestimate the impact of the Internet on data warehousing. The Internet helped remove the mystique of the computer. Executives use the Internet in their daily lives and are no longer wary of touching the keyboard. The end-user tool vendors recognized the impact of the Internet, and most of them seized upon that realization: to design their interface such that it replicated some of the look-and-feel features of the popular Internet browsers and search engines. The sophistication—and simplicity—of these tools has led to a widespread use of BI by business analysts and executives.

Another important event taking place in the last few years is the transformation from technology chasing the business to the business demanding technology. In the early days of BI, the information technology (IT) group recognized its value and tried to sell its merits to the business community. In some unfortunate cases, the IT folks set out to build a data warehouse with the hope that the business community would use it. Today, the value of a sophisticated decision support environment is widely recognized throughout the business. As an example, an effective customer relationship management program could not exist without strategic (data warehouse with associated marts) and a tactical (operational data store and oper mart) decision-making capabilities. (See Figure 1.1)

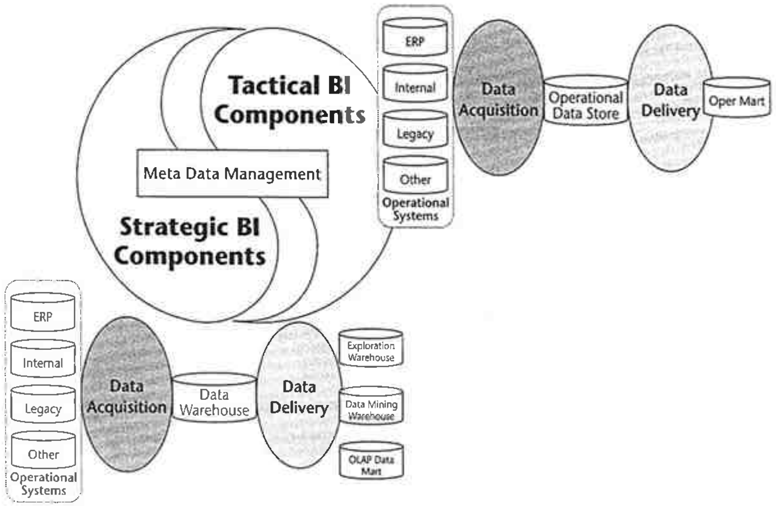


Figure 1.1 Strategic and tactical portions of a BI environment.

BI Architecture

One of the most significant developments during the last 10 years has been the introduction of a widely accepted architecture to support all BI technological demands. This architecture recognized that the EIS approach had several major flaws, the most significant of which was that the EIS data structures were often fed directly from source systems, resulting in a very complex data acquisition environment that required significant human and computer resources to maintain. The Corporate Information Factory (CIF) (see Figure 1.2), the architecture used in most decision support environments today, addressed that deficiency by segregating data into five major databases (operational systems, data warehouse, operational data store, data marts, and oper marts) and incorporating processes to effectively and efficiently move data from the source systems to the business users.

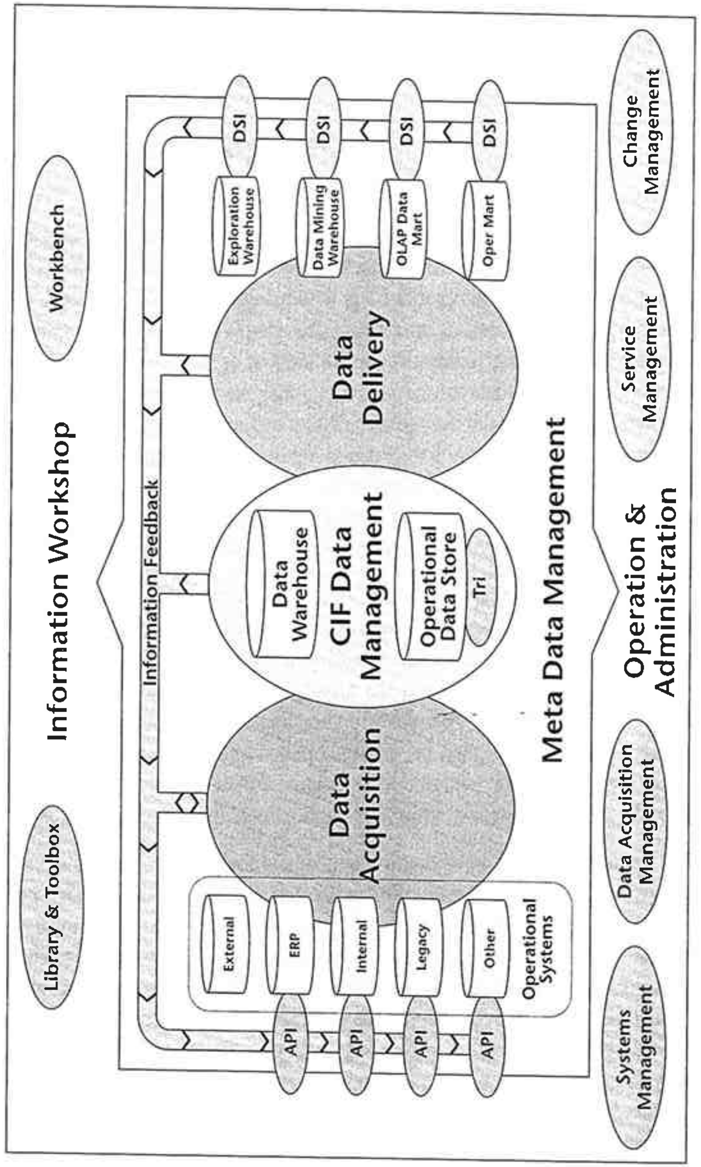


Figure 1.2 The Corporate Information Factory.

These components were further separated into two major groupings of components and processes:

- *Getting data in* consists of the processes and databases involved in acquiring data from the operational systems, integrating it, cleaning it up, and putting it into a database for easy usage. The components of the CIF that are found in this function:
  - The operational system databases (source systems) contain the data used to run the day-to-day business of the company. These are still the major source of data for the decision support environment.
  - The data warehouse is a collection or repository of integrated, detailed, historical data to support strategic decision-making.
  - The operational data store is a collection of integrated, detailed, current data to support tactical decision making.
  - Data acquisition is a set of processes and programs that extracts data for the data warehouse and operational data store from the operational systems. The data acquisition programs perform the cleansing as well as the integration of the data and transformation into an enterprise format. This enterprise format reflects an integrated set of enterprise business rules that usually causes the data acquisition layer to be the most complex component in the CIF. In addition to programs that transform and clean up data, the data acquisition layer also includes audit and control processes and programs to ensure the integrity of the data as it enters the data warehouse or operational data store.
- *Getting information out* consists of the processes and databases involved in delivering BI to the ultimate business consumer or analyst. The components of the CIF that are found in this function:
  - The data marts are derivatives from the data warehouse used to provide the business community with access to various types of strategic analysis.
  - The oper marts are derivatives of the ODS used to provide the business community with dimensional access to current operational data.
  - Data delivery is the process that moves data from the data warehouse into data and oper marts. Like the data acquisition layer, it manipulates the data as it moves it. In the case of data delivery, however, the origin is the data warehouse or ODS, which already contains high-quality, integrated data that conforms to the enterprise business rules.

The CIF didn't just happen. In the beginning, it consisted of the data warehouse and sets of lightly summarized and highly summarized data—initially a collection of the historical data needed to support strategic decisions. Over time, it spawned the operational data store with a focus on the tactical decision support requirements as well. The lightly and highly summarized sets of data evolved into what we now know are data marts.

Let's look at the CIF in action. Customer Relationship Management (CRM) is a highly popular initiative that needs the components for tactical information (operational systems, operational data store, and oper marts) and for strategic information (data warehouse and various types of data marts). Certainly this technology is necessary for CRM, but CRM requires more than just the technology—it also requires alignment of the business strategy, corporate culture and organization, and customer information in addition to technology to provide long-term value to both the customer and the organization. An architecture such as that provided by the CIF fits very well within the CRM environment, and each component has a specific design and function within this architecture. We describe each component in more detail later in this chapter.

CRM is a popular application of the data warehouse and operational data store but there are many other applications. For example, the enterprise resource planning (ERP) vendors such as SAP, Oracle, and PeopleSoft have embraced data warehousing and augmented their tool suites to provide the needed capabilities. Many software vendors are now offering various plug-ins containing generic analytical applications such as profitability or key performance indicator (KPI) analyses. We will cover the components of the CIF in far greater detail in the following sections of this chapter.

The evolution of data warehousing has been critical in helping companies better serve their customers and improve their profitability. It took a combination of technological changes and a sustainable architecture. The tools for building this environment have certainly come a long way. They are quite sophisticated and offer great benefit in the design, implementation, maintenance, and access to critical corporate data. The CIF architecture capitalizes on these technology and tool innovations. It creates an environment that segregates data into five distinct stores, each of which has a key role in providing the business community with the right information at the right time, in the right place, and in the right form. So, if you're a data warehousing late majority or even a laggard, take heart. It was worth the wait.

What Is a Data Warehouse?

Before we get started with the actual description of the modeling techniques, we need to make sure that all of us are on the same page in terms of what we mean by a data warehouse, its role and purpose in BI, and the architectural components that support its construction and usage.

Role and Purpose of the Data Warehouse

As we see in the first section of this chapter, the overall BI architecture has evolved considerably over the past decade. From simple reporting and EIS systems to multidimensional analyses to statistical and data mining requirements to exploration capabilities, and now the introduction of customizable analytical applications, these technologies are part of a robust and mature BI environment. See Figure 1.3 for the general timeframe for each of these technological advances.

Given these important but significantly different technologies and data format requirements, it should be obvious that a repository of quality, trusted data in a flexible, reusable format must be the starting point to support and maintain any BI environment. The data warehouse has been a part of the BI architecture from the very beginning. Different methodologies and data warehouse gurus have given this component various names such as:

A staging area. A variation on the data warehouse is the "back office" staging area where data from the operational systems is first brought together. It is an informally designed and maintained grouping of data whose only purpose is to feed multidimensional data marts.

The information warehouse. This was an early name for the data warehouse used by IBM and other vendors. It was not as clearly defined as the staging area and, in many cases, encompassed not only the repository of historical data but also the various data marts in its definition.

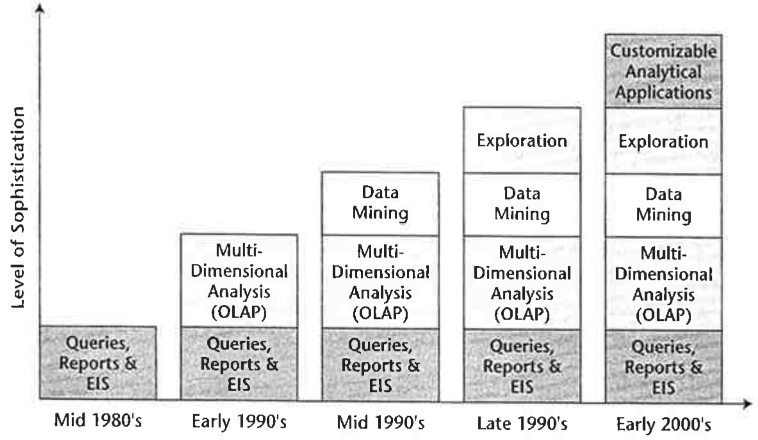


Figure 1.3 Evolving BI technologies.

The data warehouse environment must align varying skill sets, functionality, and technologies. Therefore it must be designed with two ideas in mind. First, it must be at the proper level of grain, or detail, to satisfy all the data marts. That is, it must contain the least common denominator of detailed data to supply aggregated, summarized marts as well as transaction-level exploration and mining warehouses.

Second, its design must not compromise the ability to use the various technologies for the data marts. The design must accommodate multidimensional marts as well as statistical, mining, and exploration warehouses. In addition, it must accommodate the new analytical applications being offered and be prepared to support any new technology coming down the pike. Thus the schemas it must support consist of star schemas, flat files, statistical subsets of normalized data, and whatever the future brings to BI. Given these goals, let's look at how the data warehouse fits into a comprehensive architecture supporting this mature BI environment.

The Corporate Information Factory

The Corporate Information Factory (CIF) is a widely accepted conceptual architecture that describes and categorizes the information stores used to operate and manage a successful and robust BI infrastructure. These information stores support three high-level organizational processes:

- *Business operations* are concerned with the ongoing day-to-day operations of the business. It is within this function that we find the operational transaction-processing systems and external data. These systems help run the business, and they are usually highly automated. The processes that support this function are fairly static, and they change only in quantum leaps. That is, the operational processes remain constant from day to day, and only change through a conscious effort by the company.
- *Business intelligence* is concerned with the ongoing search for a better understanding of the company, of its products, and of its customers. Whereas business operations processes are static, business intelligence includes processes that are constantly evolving, in addition to static processes. These processes can change as business analysts and knowledge workers explore the information available to them, using that information to help them develop new products, measure customer retention, evaluate potential new markets, and perform countless other tasks. The business intelligence function supports the organization's strategic decision-making process.

- *Business management* is the function in which the knowledge and new insights developed in business intelligence are institutionalized and introduced into the daily business operations throughout the enterprise. Business management encompasses the tactical decisions that an organization makes as it carries out its strategies.

Taken as a whole, the CIF can be used to identify all of the information management activities that an organization conducts. The operational systems continue to be the backbone of the enterprise, running the day-to-day business. The data warehouse collects the integrated, historical data supporting customer analysis and segmentation, and the data marts provide the business community with the capabilities to perform these analyses. The operational data store and associated oper marts support the near-real-time capture of integrated customer information and the management of actions to provide personalized customer service.

Let's examine each component of the CIF in a bit more detail.

Operational Systems

Operational systems are the ones supporting the day-to-day activities of the enterprise. They are focused on processing transactions, ranging from order entry to billing to human resources transactions. In a typical organization, the operational systems use a wide variety of technologies and architectures, and they may include some vendor-packaged systems in addition to in-house custom-developed software. Operational systems are static by nature; they change only in response to an intentional change in business policies or processes, or for technical reasons, such as system maintenance or performance tuning.

These operational systems are the source of most of the electronically maintained data within the CIF. Because these systems support time-sensitive real-time transaction processing, they have usually been optimized for performance and transaction throughput. Data in the operational systems environment may be duplicated across several systems, and is often not synchronized. These operational systems represent the first application of business rules to an organization's data, and the quality of data in the operational systems has a direct impact on the quality of all other information used in the organization.

Data Acquisition

Many companies are tempted to skip the crucial step of truly integrating their data, choosing instead to deploy a series of uncoordinated, unintegrated data marts. But without the single set of business rule transformations that the data acquisition layer contains, these companies end up building isolated, user- or department-specific data marts. These marts often cannot be combined to produce valid information, and cannot be shared across the enterprise. The net effect of skipping a single, integrated data acquisition layer is to foster the uncontrolled proliferation of silos of analytical data.

Data Warehouse

The universally accepted definition of a data warehouse developed by Bill Inmon in the 1980s is "a subject-oriented, integrated, time variant and non-volatile collection of data used in strategic decision making"[1]. The data warehouse acts as the central point of data integration—the first step toward turning data into information. Due to this enterprise focus, it serves the following purposes.

First, it delivers a common view of enterprise data, regardless of how it may later be used by the consumers. Since it is the common view of data for the business consumers, it supports the flexibility in how the data is later interpreted (analyzed). The data warehouse produces a stable source of historical information that is constant, consistent, and reliable for any consumer.

Second, because the enterprise as a whole has an enormous need for historical information, the data warehouse can grow to huge proportions (20 to 100 terabytes or more!). The design is set up from the beginning to accommodate the growth of this information in the most efficient manner using the enterprise's business rules for use throughout the enterprise.

Finally, the data warehouse is set up to supply data for any form of analytical technology within the business community. That is, many data marts can be created from the data contained in the data warehouse rather than each data mart serving as its own producer and consumer of data.

Operational Data Store

The operational data store (ODS) is used for tactical decision making, whereas the data warehouse supports strategic decisions. It has some characteristics that are similar to those of the data warehouse but is dramatically different in other aspects:

- It is subject oriented like a data warehouse.
- Its data is fully integrated like a data warehouse.

[1] Building the Data Warehouse, Third Edition by W.H. Inmon, Wiley Publishing, Inc., 2001.

- Its data is current—or as current as technology will allow. This is a significant difference from the historical nature of the data warehouse. The ODS has minimal history and shows the state of the entity as close to real time as feasible.
- Its data is volatile or updatable. This too is a significant departure from the static data warehouse. The ODS is like a transaction-processing system in that, when new data flows into the ODS, the fields affected are overwritten or updated with the new information. Other than an audit trail, no history of the previous contents is retained.
- Its data is almost entirely detailed with a small amount of dynamic aggregation or summarization. The ODS is most often designed to contain the transaction-level data, that is, the lowest level of detail for the subject area.

The ODS is the source of near-real-time, accurate, integrated data about customers, products, inventory, and so on. It is accessible from anywhere in the corporation and is not application specific. There are four classes of ODS commonly used; each has distinct characteristics and usage, but the most significant difference among them is the frequency of updating, ranging from daily to almost real time (subminute latency). Unlike a data warehouse, in which very little reporting is done against the warehouse itself (reporting is pushed out to the data marts), business users frequently access an ODS directly.

Data Delivery

Data delivery is generally limited to operations such as aggregation of data, filtering by specific dimensions or business requirements, reformatting data to ease end-user access or to support specific BI access software tools, and finally delivery or transmittal of data across the organization. The data delivery infrastructure remains fairly static in a mature CIF environment; however, the data requirements of the data marts evolve rapidly to keep pace with changing business information needs. This means that the data delivery layer must be flexible enough to keep pace with these demands.

Data Marts

Data marts are a subset of data warehouse data and are where most of the analytical activities in the BI environment take place. The data in each data mart is usually tailored for a particular capability or function, such as product profitability analysis, KPI analyses, customer demographic analyses, and so on. Each specific data mart is not necessarily valid for other uses. All varieties of data marts have universal and unique characteristics. The universal ones are that they contain a subset of data warehouse data, they may be physically co-located with the data warehouse or on their own separate platform, and they

range in size from a few megabytes to multiple gigabytes to terabytes! To maximize your data warehousing ROI, you need to embrace and implement data warehouse architectures that enable this full spectrum of analysis.

Meta Data Management

Meta data management is the set of processes the collect, manage, and deploy meta data throughout the CIF. The scope of meta data managed by these processes includes three categories. *Technical* meta data describes the physical structures in the CIF and the detailed processes that move and transform data in the environment. *Business* meta data describes the data structures, data elements, business rules, and business usage of data in the CIF. Finally, *Administrative* meta data describes the operation of the CIF, including audit trails, performance metrics, data quality metrics, and other statistical meta data.

Information Feedback

Information feedback is the sharing mechanism that allows intelligence and knowledge gathered through the usage of the Corporate Information Factory to be shared with other data stores, as appropriate. It is the use of information feedback that identifies an organization as a true "learning organization." Examples of information feedback include:

- Pulling derived measures such as new budget targets from data marts and feeding them back to the data warehouse where they will be stored for historical analysis.
- Transmitting data that has been updated in an operational data store (through the use of a Transactional Interface) to appropriate operational systems, so that those data stores can reflect the new data.
- Feeding the results of analyses, such as a customer's segment classification and life time value score, back to the operational systems or ODS.

Information Workshop

The information workshop is the set of tools available to business users to help them use the resources of the Corporate Information Factory. The information workshop typically provides a way to organize and categorize the data and other resources in the CIF, so that users can find and use those resources. This is the mechanism that promotes the sharing and reuse of analysis across the organization. In some companies, this concept is manifested as an intranet portal, which organizes information resources and puts them at business users' fingertips. We classify the components of the information workshop as the library, toolbox, and workbench.

The library and toolbox usually represent the organization's first attempts to create an information workshop. The library component provides a directory of the resources and data available in the CIF, organized in a way that makes sense to business users. This directory is much like a library, in that there is a standard taxonomy for categorizing and ordering information components. This taxonomy is often based on organizational structures or high-level business processes. The toolbox is the collection of reusable components (for example, analytical reports) that business users can share, in order to leverage work and analysis performed by others in the enterprise. Together, these two concepts constitute a basic version of the information workshop capability.

More mature CIF organizations support the information workshop concept through the use of integrated information workbenches. In the workbench, meta data, data, and analysis tools are organized around business functions and tasks. The workbench dispenses with the rigid taxonomy of the library and toolbox, and replaces it with a task-oriented or workflow interface that supports business users in their jobs.

Operations and Administration

Operation and administration include the crucial support and infrastructure functions that are necessary for a growing, sustainable Corporate Information Factory. In early CIF implementations, many companies did not recognize how important these functions were, and they were often left out during CIF planning and development. The operation and administration functions include CIF Data Management, Systems Management, Data Acquisition Management, Service Management, and Change Management. Each of these functions contains a set of procedures and policies for maintaining and enhancing these critically important processes.

The Multipurpose Nature of the Data Warehouse

Hopefully by now, you have a good understanding of the role the data warehouse plays in your BI environment. It not only serves as the integration point for your operational data, it must also serve as the distribution point of this data into the hands of the various business users. If the data warehouse is to act as a stable and permanent repository of historical data for use in your strategic BI applications, it should have the following characteristics:

It should be enterprise focused. The data warehouse should be the starting point for all data marts and analytical applications; thus, it will be used by multiple departments, maybe even multiple companies or subdivisions.

A difficult but mandatory part of any data warehouse design team's activities must be the resolution of conflicting data elements and definitions. The participation by the business community is also obligatory.

Its design should be as resilient to change as possible. Since the data warehouse is used to store massive, detailed, strategic data over multiple years, it is very undesirable to unload the data, redesign the database, and then reload the data. To avoid this, you should think in terms of a process-independent, application-independent, and BI technology-independent data model. The goal is to create a data model that can easily accommodate new data elements as they are discovered and needed without having to redesign the existing data elements or data model.

It should be designed to load massive amounts of data in very short amounts of time. The data warehouse database design must be created with a minimum of redundancy or duplicated attributes or entities. Most databases have bulk load utilities that include a range of features and functions that can help optimize this process. These include parallelization options, loading data by block, and native application program interfaces (APIs). They may mean that you must turn off indexing, and they may require flat files. However, it is important to note that a poorly or ineffectively designed database cannot be overcome even with the best load utilities.

It should be designed for optimal data extraction processing by the data delivery programs. Remember that the ultimate goal for the data warehouse is to feed the plethora of data marts that are then used by the business community. Therefore, the data warehouse must be well documented so that data delivery teams can easily create their data delivery programs. The quality of the data, its lineage, any calculations or derivations, and its meaning should all be clearly documented.

Its data should be in a format that supports any and all possible BI analyses in any and all technologies. It should contain the least common denominator level of detailed data in a format that supports all manner of BI technologies. And it must be designed without bias or any particular department's utilization only in mind.

Types of Data Marts Supported

Today, we have a plethora of technologies supporting different analytical needs—Online Analytical Processing (OLAP), exploration, data mining and statistical data marts, and now customizable analytical applications. The unique characteristics come from the specificity of the technology supporting each type of data mart:

OLAP data mart. These data marts are designed to support generalized multidimensional analysis, using OLAP software tools. The data mart is designed using the star schema technique or proprietary "hypercube" technology. The star schema or multidimensional database management system (MD DBMS) is great for supporting multidimensional analysis in data marts that have known, stable requirements, fairly predictable queries with reasonable response times, and recurring reports. These analyses may include sales analysis, product profitability analysis, human resources headcount distribution tracking, or channel sales analysis.

Exploration warehouse. While most common data marts are designed to support specific types of analysis and reporting, the exploration warehouse is built to provide exploratory or true "ad hoc" navigation through data. After the business explorers make a useful discovery, that analysis may be formalized through the creation of another form of data mart (such as an OLAP one), so that others may benefit from it over time. New technologies have greatly improved the ability to explore data and to create a prototype quickly and efficiently. These include token, encoded vector, and bitmap technologies.

Data-mining or statistical warehouse. The data-mining or statistical warehouse is a specialized data mart designed to give researchers and analysts the ability to delve into the known and unknown relationships of data and events without having preconceived notions of those relationships. It is a safe haven for people to perform queries and apply mining and statistical algorithms to data, without having to worry about disabling the production data warehouse or receiving biased data such as that contained in multidimensional designs (in which only known, documented relationships are constructed).

Customizable analytical applications. These new additions permit inexpensive and effective customization of generic applications. These "canned" applications meet a high percentage of every company's generic needs yet can be customized for the remaining specific functionality. They require that you think in terms of variety and customization through flexibility and quick responsiveness.

Types of BI Technologies Supported

The reality is that database structures for data marts vary across a spectrum from normalized to denormalized to flat files of transactions. The ideal situation

is to craft the data mart schemas *after* the requirements are established. Unfortunately, the database structure/solution is often selected *before* the specific business needs are known. Those of us in the data warehouse consulting business have witnessed development teams debating star versus normalized designs before even starting business analysis. For whatever reason, architects and data modelers latch onto a particular design technique—perhaps through comfort with a particular technique or ignorance of other techniques—and force all data marts to have that one type of design. This is similar to the person who is an expert with a hammer—everything he or she sees resembles a nail.

Our recommendation for data mart designs is that the schemas should be based on the usage of the data and the type of information requested. There are no absolutes, of course, but we feel that the best design to support all the types of data marts will be one that does not preestablish or predetermine the data relationships. An important caveat here is that the data warehouse that feeds the marts will be required to support any and all forms of analysis—not just multidimensional forms.

To determine the best database design for your business requirements and ensuing data mart, we recommend that you develop a simple matrix that plots the volatility of the data against a type of database design required, similar to the one in Figure 1.4. Such a matrix allows designers, architects, and database administrators (DBAs) to view where the overall requirements lie in terms of the physical database drivers, that is, volatility, latency, multiple subject areas, and so on, and the analytical vehicle that will supply the information (via the scenarios that were developed), for example, repetitive delivery, ad hoc reports, production reports, algorithmic analysis, and so on.

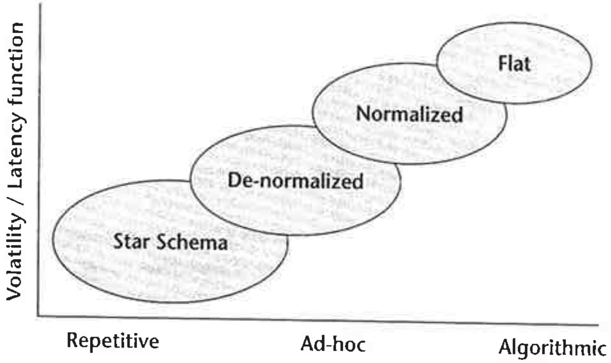


Figure 1.4 Business requirements–data mart design matrix.

Characteristics of a Maintainable Data Warehouse Environment

With this as a background, what does a solid, maintainable data warehouse data model look like? What are the characteristics that should be considered when designing any data warehouse, whether for a company just beginning its BI initiative or for a company having a sophisticated set of technologies and users, whether the company has only one BI access tool today or has a plethora of BI technologies available?

The methodology for building a BI environment is iterative in nature. We are fortunate today to have many excellent books devoted to describing this methodology. (See the "Recommended Reading" section at the end of this book.) In a nutshell, here are the steps:

1. First, select and document the business problem to be solved with a business intelligence capability (data mart of some sort).
2. Gather as many of the requirements as you can. These will be further refined in the next step.
3. Determine the appropriate end-user technology to support the solution (OLAP, mining, exploration, analytical application, and so on).
4. Build a prototype of the data mart to test its functionality with the business users, redesigning it as necessary.
5. Develop the data warehouse data model, based on the user requirements and the business data model.
6. Map the data mart requirements to the data warehouse data model and ultimately back to the operational systems, themselves.
7. Generate the code to perform the ETL and data delivery processes. Be sure to include error detection and correction and audit trail procedures in these processes.
8. Test the data warehouse and data mart creation processes. Measure the data quality parameters and create the appropriate meta data for the environment.
9. Upon acceptance, move the first iteration of the data warehouse and the data mart into production, train the rest of the business community, and start planning for the next iteration.

WARNING

Nowhere do we recommend that you build an entire data warehouse containing all the strategic enterprise data you will ever need before building the first analytical capability (data mart). Each successive business problem solved by another data mart implementation will add the growing set of data serving as the foundation in your data warehouse. Eventually, the amount of data that must be added to the data warehouse to support a new data mart will be negligible because most of it will already be present in the data warehouse.

Since you will not know how large the data warehouse will ultimately be, nor do you know all of the BI technologies that will eventually be brought to bear upon strategic problems in your enterprise, you must make some educated assumptions and plan accordingly. You can assume that the warehouse will become one of the largest databases found in your enterprise. It is not unusual for the data warehouse size to start out in the low gigabyte range and then grow fairly rapidly to hundreds of gigabytes, terabytes, and some now predict pedabytes. So, regardless of where you are in your BI life cycle—just starting or several years into building the environment—the relational databases are still the best choice for your database management system (DBMS). They have the advantage of being very conducive to nonredundant, efficient database design. In addition, their deployment for the data warehouse means you can use all the sophisticated and useful characteristics of a relational DBMS:

- **Access to the data by most any tool (data modeling, ETL, meta data, and BI access).** All use SQL on the relational database.
- **Scalability in terms of the size of data being stored.** The relational databases are still superior in terms of storing massive amounts of data.
- **Parallelism for efficient and extremely fast processing of data.** The relational databases excel at this function.
- **Utilities such as bulk loaders, defragmentation, and reorganization capabilities, performance monitors, backup and recovery functions, and index wizards.** Again, the relational databases are ideal for supporting a repository of strategic data.

There may come a time when the proprietary multidimensional databases (MOLAP) can effectively compete with their relational cousins, but that is not the situation currently.

The Data Warehouse Data Model

Given that we recommend a relational DBMS for your data warehouse, what should the characteristics of the data model for that structure look like? Again, let's look at some assumptions before going into the characteristics of the model:

- The data warehouse is assumed to have an enterprise focus at its heart. This means that the data contained in it does not have a bias toward one department or one part of the enterprise over another. Therefore, the ultimate BI capabilities may require further processing (for example, the use of a data mart) to "customize" them for a specific group, but the starting material (data) can be used by all.
- As a corollary to the above assumption, it is assumed that the data within data warehouse does not violate any business rules established by the enterprise. The data model for the data warehouse must demonstrate adherence to these underlying rules through its form and documentation.
- The data warehouse must be loaded with new data as quickly and efficiently as possible. Batch windows, if they exist at all, are becoming smaller and smaller. The bulk of the work to get data into a data warehouse must occur in the ETL process, leaving minimal time to load the data.
- The data warehouse must be set up from the beginning to support multiple BI technologies—even if they are not known at the time of the first data mart project. Biasing the data warehouse toward one technology, such as multidimensional analyses, effectively eliminates the ability to satisfy other needs such as mining and statistical analyses.
- The data warehouse must gracefully accommodate change in its data and data structures. Given that we do not have all of the requirements or known uses of the strategic data in the warehouse from the very beginning, we can be assured that changes will happen as we build onto the existing data warehouse foundation.

With these assumptions in mind, let's look at the characteristics of the ideal data warehouse data model.

Nonredundant

To accommodate the limited load cycles and the massive amount of data that most data warehouses must have, the data model for the data warehouse should contain a minimum amount of redundancy. Redundancy adds a tremendous burden to the load utilities and to the designers who must worry about ensuring that all redundant data elements and entities get the correct data at the correct time. The more redundancy you introduce to your data warehouse data model, the more complex you make the ultimate process of "getting data in."

This does not mean that redundancy is not ever found in the data warehouse. In Chapter 4, we describe when and why some redundancy is introduced into the data warehouse. The key though is that redundancy is controlled and managed with forethought.

Stable

As mentioned earlier, we build the data warehouse in an iterative fashion, which has the benefit of getting a data mart created quickly but runs the risk of missing or misstating significant business rules or data elements. These would be determined or highlighted as more and more data marts came online. It is inevitable that change will happen to the data warehouse and its data model.

It is well known that what changes most often in any enterprise are its processes, applications, and technology. If we create a data model dependent upon any of these three factors, we can be assured of a major overhaul when one of the three changes. Therefore, as designers, we must use a data-modeling technique that mitigates this problem as much as possible yet captures the all-important business rules of the enterprise. The best data-modeling technique for this mitigation is to create a process-, application-, and technology-independent data model.

On the other hand, since change is inevitable, we must be prepared to accommodate newly discovered entities or attributes as new BI capabilities and data marts are created. Again, the designer of the data warehouse must use a modeling technique that can easily incorporate a new change without someone's having to redesign the existing elements and entities already implemented. This model is called a system model, and will be described in Chapter 3 in more detail.

Consistent

Perhaps the most essential characteristic of any data warehouse data model is the consistency it brings to the business for its most important asset—its data. The data models contain all the meta data (definitions, physical characteristics, aliases, business rules, data owners and stewards, domains, roles, and so on) that is critically important to the ultimate understanding of the business users of what they are analyzing. The data model creation process must reconcile outstanding issues, data discrepancies, and conflicts before any ETL processing or data mapping can occur.

Flexible in Terms of the Ultimate Data Usage

The single most important purpose for the data warehouse is to serve as a solid, reliable, consistent foundation of data for any and all BI capabilities. It should be clear by now that, regardless of what your first BI capability is, you must be able to serve all business requirements regardless of their technologies. Therefore, the data warehouse data model must remain application and technology independent, thus making it ideal to support any application or technology.

On the other hand, the model must uphold the business rules established for the organization, and that means that the data model must be more than simply flat files. Flat files, while a useful base to create star schemas, data mining, and exploration subsets of data, do not enforce, or even document, any known business rules. As the designer, you must go one step further and create a real data model with the real business rules, domains, cardinalities, and optionalities specified. Otherwise, subsequent usage of the data could be mishandled, and violations in business rules could occur.

The Codd and Date Premise

Given all of the above characteristics of a good data warehouse data model, we submit that the best data-modeling technique you can use is one based on the original relational database design—the entity-relationship diagram (ERD) developed by Chris Date and Ted Codd. The ERD is a proven and reliable data-modeling approach with straightforward rules of construction. The normalization rules discussed in Chapter 3 yield a stable, consistent data model that upholds the policies and rules of engagement established by the enterprise, while lending a tremendous amount of flexibility in how the data is later analyzed by the data marts. The resulting database is the most efficient in terms of storage and data loading as well. It is, however, not perfect, as we will see in the next section.

While we certainly feel that this approach is elegant in the extreme, more importantly, this data-modeling technique upholds all of the features and characteristics we specified for a sustainable, flexible, maintainable, and understandable data warehouse environment.

The resultant data model for your data warehouse is translatable, using any technology, into a database design that is:

Reliable across the business. It contains no contradictions in the way that data elements or entities are named, related to each other, or documented.

Sharable across the enterprise. The data warehouse resulting from the implementation of this data model can be accessed by multiple data delivery processes and users from anywhere in the enterprise

Flexible in the types of data marts it supports. The resulting database will not bias your BI environment in one direction or another. All technological opportunities will still be available to you and your enterprise.

Correct across the business. The data warehouse data model will provide an accurate and faithful representation of the way information is used in the business.

Adaptable to changes. The resulting database will be able to accommodate new elements and entities, while maintaining the integrity of the implemented ones.

Impact on Data Mart Creation

Now that we have described the characteristics of a solid data warehouse data model and have recommended an ERD or normalized (in the sense of Date and Codd) approach, let's look at the ramifications that decision will have on our overall BI environment.

The most common applications that use the data warehouse data are multidimensional ones—at least today. The dimensions used in the star schemas correlate roughly to the subject areas developed in the subject area model—order, customer, product, market segment—and time. To answer the questions, "How many orders for what products did we get in the Northeast section from January to June this year?" would take a significant amount of effort if we were to use the data warehouse as the source of data for that query. It would require a rather large join across several big entities (Order, Order Line Item, Product, Market Segment, with the restriction of the timeframe in the SQL statement). This is not a pretty or particularly welcomed situation for the average business user who is distantly familiar with SQL.

So, what we can see about this situation is that data warehouse access will have to be restricted and used by only those business users who are very sophisticated in database design and SQL. If an enterprise has good exploration and mining technology, it may choose to cut off all access to the data warehouse, thus requiring all business users to access an OLAP mart, or exploration or data mining warehouse instead.

Is this a problem? Not really. All BI environments must have "back room" capabilities of one sort or another. It is in the back room that we perform the difficult tasks of integration, data hygiene, error correction and detection, transformation, and the audit and control mechanisms to ensure the quality of the strategic data anyway. Therefore, all BI environments have this "closed off to the public" section of their environment. We have simply taken it one step further and said that this section should be formally modeled, created, and maintained

In the data-mart-only world, the data delivery processes, described earlier, must take on not only the burden of ensuring the proper delivery of the information to the right mart at the right time but must also take on the entire set of ETL tasks found in the data acquisition processing over and over again. Given this situation, it should be obvious that the data delivery processes can be simplified greatly if all they have to worry about is extracting the data they specifically need from a consistent, quality source (the data warehouse), format it into that required by the data mart technology (star schema, flat file, normalized subset, and so on), and deliver the data to the data mart environment for uploading.

As another benefit to constructing the data warehouse from a solid, ERD-based data model, you get a very nice set of reusable data entities and elements. In a data-mart-only environment, each mart must carry all the detailed data it requires within its database. Unless the two data marts share common conformed dimensions, integrating the two may be difficult, or even impossible. Imagine if a repository of detailed data existed that the data delivery processes could extract from and the BI access tools could access, if they needed to, at any time without having to replicate the data over and over! That is another significant benefit the data warehouse brings to your BI environment.

Summary

There are several BI methodologies and consultants who will tell you that you do not need a data warehouse, that the combination of all the data marts together creates the "data warehouse," or at least a virtual one, or that really, all the business really wants is just a standalone data mart. We find all of these approaches to be seriously lacking in sustainability and sophistication. This book takes a "best practices" approach to creating a data warehouse. The best practices we use are a set of recommendations that tells designers what actions they should take or avoid, thus maximizing the success of their overall efforts. These recommendations are based on the years of experience in the field, participation in many data warehouse projects, and the observation of many successful and maintainable data warehouse environments. Clearly, no one method is perfect, nor should one be followed blindly without thought being given to the specific situation. You should understand what works best in your environment and then apply these rules as you see fit, altering them as changes and new situations arise.

In spite of this caveat, this book is filled with useful and valuable information, guidelines, and hints. In the following chapters, we will describe the data models needed in more detail, go over the construction of the data warehouse data model step by step, and discuss deployment issues and problems you may encounter along the way to creating a sustainable and maintainable business intelligence environment. By the end of the book, you should be fully qualified to begin constructing your BI environment armed with the best design techniques possible for your data warehouse.

CHAPTER 13

Comparison of Data Warehouse Methodologies

It's appropriate at this point to recognize that the Corporate Information Factory (CIF) is not the only business intelligence (BI) architecture. Another architecture worth noting is Dr. Ralph Kimball's multidimensional (MD) architecture[1]. This chapter starts with a brief description of the MD and CIF architectures and then highlights the significant similarities and differences between the two by using the criteria of scope, perspective, data flow, implementation speed and cost, volatility, complexity, and functionality.

Perhaps as a way of introducing the two architectures, we should explain that we believe that a combination of the data-modeling techniques found in the two architectural approaches works best—ERD or normalization techniques for the data warehouse and the star schema data model for multidimensional data marts. That said, it is important that BI architects study their situation, politics, and culture to determine what works best in their environment.

The Multidimensional Architecture

The MD architecture (see Figure 13.1) is based on the premise that all BI analyses have at their foundation a multidimensional design. The star schema is an elegant data model that layers multidimensional meta data over what is basically a

[1] See *The Data Warehouse Lifecycle Toolkit*, Ralph Kimball et al., Wiley Publishing, Inc., 1998.

two-dimensional data store (columns and rows), making it act to the user as if it were multidimensional. The star schema gave BI a solid and much needed push into the mainstream when it first appeared. It is still one of the most popular and useful designs for usage in strategic decision-making environments.

One of the more significant differences between the MD and CIF architectures is in the definition of the data mart. For the MD architecture, the aggregated data mart star schema is approximately the same as the data mart in the CIF architecture. The atomic-level data mart star schema contains the detailed data roughly equivalent to the content in the CIF's data warehouse. However, the design of the atomic-level data marts (star schemas) is significantly different from the design of the CIF data warehouse (denormalized ERD schema). These data-modeling differences constitute the main design differences in these two architectures.

All star schema-based data marts may or may not reside within the same database instance. A collection of these schemas in a single database instance is called the Data Warehouse Bus Architecture. Unlike the CIF, a separate and physically distinct data warehouse does not exist.

The MD architecture is divided into two groups of components and processes—the back room and front room. The back room is where the data-staging and data acquisition processes take place. Mapping to the operational systems and the technical meta data surrounding these maps is also part of the back room. It is roughly equivalent to the CIF's "Getting Data In" components with some notable exceptions. One is the lack of an ERD-based data warehouse, as mentioned, and the other is the presence of atomic and aggregated star schema data marts—both discussed later in this chapter. The latter appears in both the back and front rooms.

The data-staging area contains the conformed dimensions but it is also the place where surrogate keys are generated, maps to the operational systems are kept, current loads of operational data are stored, and any atomic data not currently used in the data marts is stored. Most of the heavy lifting performed by the ETL tools occurs here as well.

The Data Warehouse Bus Architecture consists of two types of data marts:

Atomic Data Marts. These data marts hold multidimensional data at the lowest common denominator level (lowest level of detail available throughout the environment). They may contain some aggregated data as well to improve query performance. The data is stored in a star schema data model.

Aggregated Data Marts. These data marts contain data related to a core business process such as marketing, sales, or finance. Generally, the atomic data marts supply the data to be aggregated for these data marts but that is not mandatory. It is possible to create an aggregated data mart directly from the data-staging area. As with the atomic data marts, data is stored in the aggregated data marts in star schema designs.

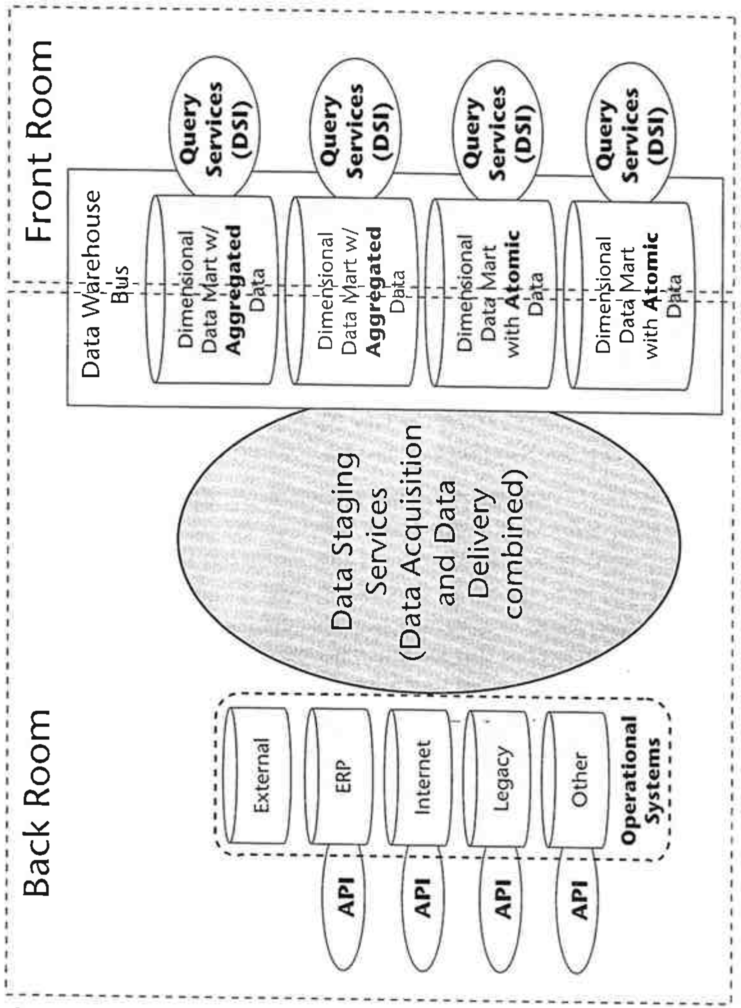

Front Room
Back Room
Data Warehouse Bus
Query Services (DSI)
Query Services (DSI)
Query Services (DSI)
Query Services (DSI)
Dimensional Data Mart w/ Aggregated Data
Dimensional Data Mart w/ Aggregated Data
Dimensional Data Mart with Atomic Data
Dimensional Data Mart with Atomic Data
Data Staging Services (Data Acquisition and Data Delivery combined)
External
ERP
Internet
Legacy
Other
Operational Systems
API
API
API
API

Figure 13.1 The multidimensional architecture.

Your need for both types of data marts depends on your business requirements and the performance of each of these structures in your environment. However, it is important to understand that the MD architecture starts and ends with its focus primarily on the individual business unit(s) or group of business users with a specific BI requirement. This singular focus is reflected in the structure of the data, which is optimized to accommodate that unit or group of users perfectly. No two star schemas are exactly alike—each provides an optimal way of accessing data for a specific set of requirements. As unit after unit or group after group is added to the list of BI recipients, either new star schemas must be built to accommodate them specifically or the existing design must be reconstructed to expand its functionality.

The front room is the interface for the business community. We see it as roughly equivalent to the CIF's "Getting Information Out" components. It is clear that the decision support interfaces (called Access Services) and their corresponding end user access tools belong in this part of the architecture. The two types of data marts also appear in the front room as the source of data for these interfaces and tools. The basic tenet of the front room is to mask or hide the complexity going on in the back room from the business community since it is believed by these authors that users of these components neither know nor care about the significant amount of energy, time, and resources poured into creating the back room.

It is in the front room that we begin to see personal data marts (also called "spreadmarts") popping up, as well as disposable data marts (data marts created for a specific short-lived business requirement). Care should be taken in both cases to ensure that these do not supplant or replace the real data marts; otherwise, you end up with chaos again.

The end user access tools consist of OLAP engines, reporting and querying tools, and maybe even some data-mining tools. We caution the reader here that the process of building a star schema limits the usefulness of these data marts for complete and unbiased data mining and statistical analyses, as well as for exploration analyses. (See Chapter 1 for more on this.) If the data is stored in only star schemas, then it becomes impossible to find unrelated patterns or correlations in the raw data. Because the star contains only known relationships, then patterns or correlations between unrelated data sets cannot be performed.

The front room also contains the query management and activity-monitoring services. These are very useful in maintaining the appropriate performance for each data mart installation. Query management involves services such as query retargeting, aggregate awareness, and query governing. Activity monitoring captures information about the usage of these databases to determine if performance and user support are optimal.

There are many other services embedded in the front room that we do not list here. For the full set, please refer to the books by Ralph Kimball et al. Suffice it to say that much of what is captured in the CIF Operations and Administration Service Management function is also captured in parts of this architecture as well.

Because the approach is predominately a bottom-up one, it is easy to violate the corporate or enterprise business rules when constructing the star schema. If there is no insistence that top-down design work be performed, the star schemas can easily become stovepipe implementations, lacking in the ability to link together, producing inconsistent and, perhaps worse, conflicting, intelligence across the enterprise. Strong and experienced multidimensional modelers, just like experienced ERD modelers, overcome this because their experience allows them to recognize the need to do so.

In addition, over the years, the MD approach has been modified in attempts to overcome the shortcoming of the lack of an enterprise view, by ensuring that the various data mart star schemas "conform" to some enterprise standards. Conformed dimensions are one way to overcome this shortcoming. According to Kimball et al., a conformed dimension is one that means the same thing to every possible fact table to which it can be joined. Ideally, this means that a conformed dimension is identical in every star schema that uses it. Examples of these are Customer, Product, Time, and Locations dimensions.

Another workaround the shortcoming was the creation of a data-staging area (not shown in Figure 13.1). In this data store, the designer consolidates all of a dimension's attributes into a single conformed dimension to be replicated to all the requesting star schemas. It is the responsibility of the design team to create, publish, maintain, and enforce the usage of these conformed dimensions throughout all data marts. Once consolidated, the conformed dimensions are permanently stored in the data-staging area. This retrofit of an enterprise standard mitigates the possible inconsistencies and discrepancies that occur in dimensions with no enterprise consideration. The data warehouse bus design concept was developed for this purpose.

The Corporate Information Factory Architecture

Chapter 1 discusses the functions and components associated with "getting data in" and "getting information out." Figure 13.2 is a simplified version of the CIF, showing these two functions and the components and processes involved in each.

The staging area (not shown in Figure 13.2) in the CIF includes persistent tables for storing the key conversion information and other reference tables used in the data acquisition process. Replicated operational data not yet used in the warehouse may also be stored there, waiting for integration and loading into the warehouse. The staging area may or may not be separate from the data warehouse but if it is on the same platform as the warehouse, it should be in its own database instance.

In the MD architecture, the back room is completely off-limits to the business community. Unlike the data-staging area in the back room of the MD architecture, business community access to the CIF data warehouse is discouraged, but exceptions for special exploration or one-time extraction needs are permitted. Other than security restrictions that you may want to implement, there is nothing to prevent its usage since the data is completely documented, integrated, and validated. However, the data model is complicated, and the business user must understand an ERD model and how to "walk a relational database" in order to use it.

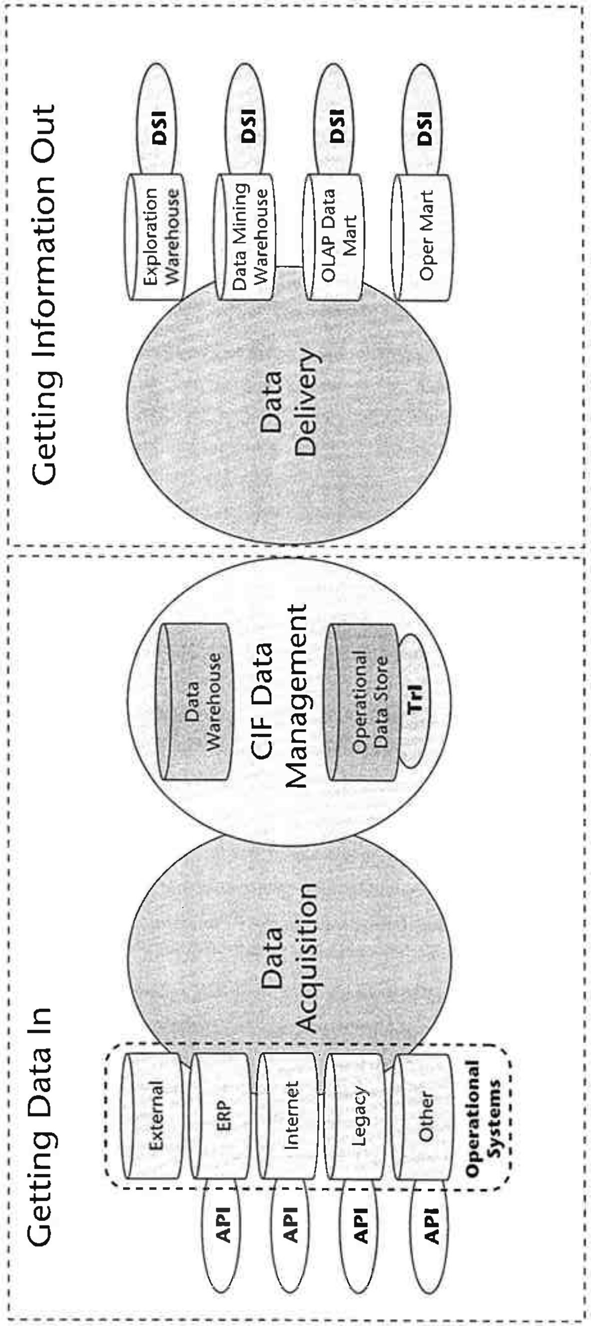


Figure 13.2 Simplified corporate information factory.

Comparison of the CIF and MD Architectures

Figure 13.3 is an adaptation of a slide from Laura Reeves of StarSoft (www.starsoftinc.com) comparing the CIF and MD architectures. The significant points in this figure are that access is generally not allowed above the diagonal line in both architectures, and there is no physical repository equivalent to the data warehouse in the MD architecture. The "data warehouse bus" shown for the MD architecture is the collection of the atomic and aggregated data marts.

Both the CIF and MD architectures have a staging area, meta data management, and sophisticated data acquisition processing. The designs of the data marts are predominantly multidimensional for both architectures, though the CIF is not limited to just this design and can support a much broader set of data mart design techniques.

What's missing in the MD architecture is a separate physical data warehouse. The "data warehouse" in this architecture as mentioned earlier is virtual and consists of the collection of all the individual data marts and their corresponding data (both atomic level and aggregated levels). The closest thing to the CIF data warehouse seems to be the "data-staging area" in the MD architecture, which, in his August 1997 *DBMS* Magazine article "A Dimensional Modeling Manifesto," Ralph Kimball states is often designed using ERD or third normal form data models.

Now, let's look more closely at the major comparison topics for the MD and CIF architectures: scope, perspective, data flow, implementation speed and cost, volatility, flexibility, functionality, and ongoing maintenance.

Scope

BI is about discovery. CIF and MD architectures both help an enterprise satisfy its basic need for more information about both itself and the environment in which it exists. CIF and MD both assume that BI requirements will emerge from business units of an organization, as well as from the organization as a whole. To illustrate how enterprise data can differ from business unit data, consider that, for a bank, "customer" might mean an individual account holder to Finance, a household of account holders to Marketing, and a non-account-holder to Customer Service. To the enterprise, "customer" means all of these and more, and distinct terms and definitions for each type of customer may be needed.

Such differences in meaning are synonymous with differences in scope. While neither of the architectures ignores enterprise scope or business unit scope, each favors one over the other. CIF places a higher priority on enterprise scope, and MD places a higher priority on business unit scope. Hence, the scope of the first few projects under the CIF architecture may be a bit larger [illegible] ID architectural project.

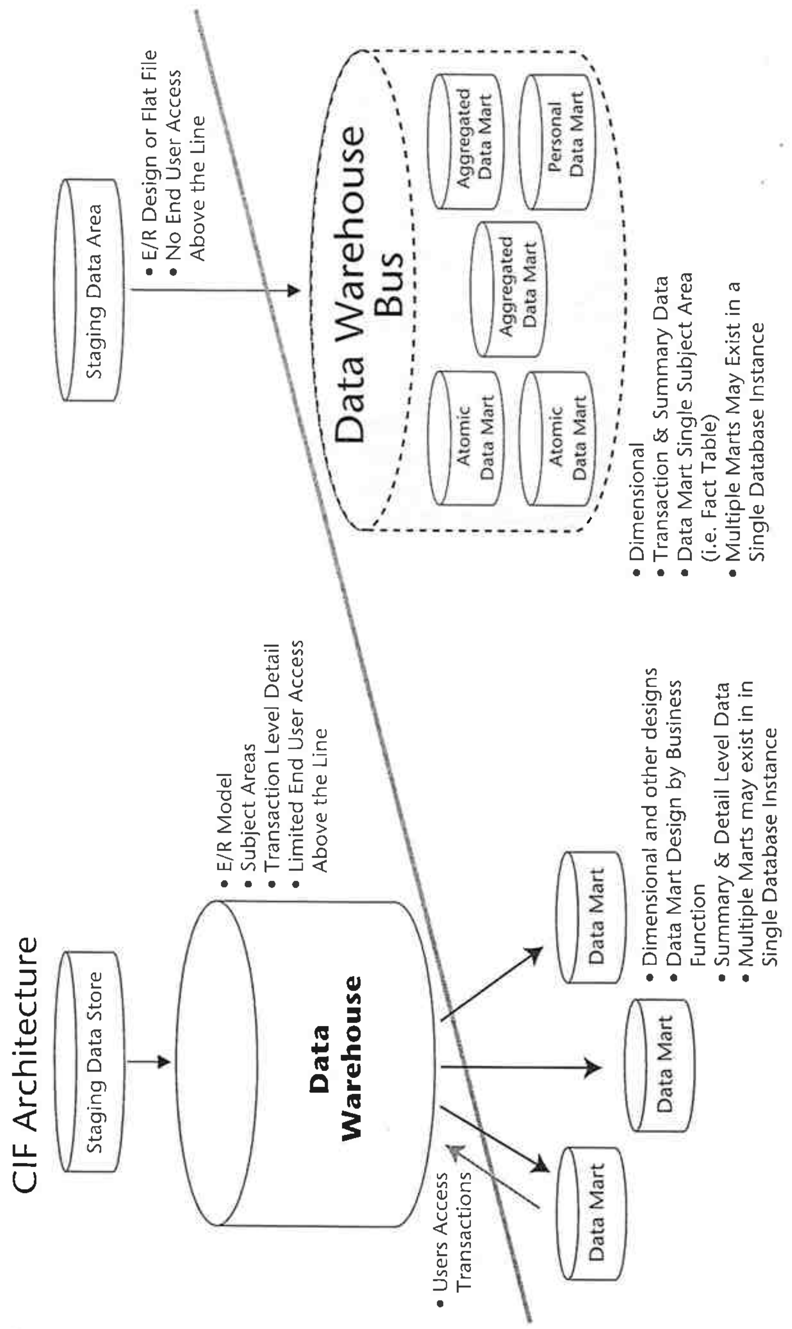


Figure 13.3 Comparison of CIF and MD architectures.

Perspective

CIF proponents frequently say that the historic problem with BI implementations is that the BI source data is difficult to locate, gather, integrate, understand, and deliver. Given an enterprise scope, they emphasize the perspective of *supplying* enterprise data. IT is often centralized and experienced at maintaining data at the enterprise level, so IT tackles the problems of supplying BI source data from an enterprise point of view. CIF proponents favor the needs of the enterprise and advocate getting the BI source data modeled for the enterprise as a prerequisite for any BI implementation. Note though, that this does not mean that the entire enterprise data must be dealt with during the first project. On the contrary, a subset of the overall enterprise's data is selected, predominantly from a subject area like Customer or Product, and the data warehouse data model and resulting database are implemented for just this small part of the overall set of enterprise data.

MD proponents frequently say the same thing about the historic problem with BI implementations. Using the same words, and given their business unit scope, they emphasize the perspective of *consuming* business unit data. Business units that consume BI data, such as Sales or Finance, are experienced with their individual needs and views. If another business unit has different needs and views, that's okay. They just don't value other business unit needs and views as much as they do their own. MD proponents favor the needs of the business unit and advocate getting the BI source data modeled for the business unit as a prerequisite for any BI implementation. It is important to note that the multidimensional modeler must strive to achieve consensus on the definition of the conformed dimensions across the enterprise, however. He or she concentrates only on those dimensions pertinent to the facts being loaded. Where a new fact is introduced that requires new dimensions not previously defined, the multidimensional modeler must again take an enterprise view and gain a consensus definition among those business areas that have some stake in that dimension.

Data Flow

To create a sustainable BI environment, one must understand the iterative nature of the projects and the relationship the ultimate environment has with the sources of data supplied to the enterprise. Like the chicken and egg paradox, BI questions create answers that create more BI questions (see Figure 13.4.). Even though BI source data starts and ends at the same places for CIF and MD, given these two architectures' unique scopes and perspectives, they view BI data flow differently. It's a matter of push versus pull. In general, the CIF approach is top-down. CIF suppliers of enterprise BI data use the business requirements to push the data from the operational systems to where it's needed. The focus is on integrating the enterprise data for usage in any data mart from the very first project.

By contrast, the MD approach is bottom-up. MD consumers of business unit BI data use the business requirements to pull the data from the operational systems to where it's needed. The focus is on getting business-unit-specific data quickly into the hands of the users with minimal regard for the overall enterprise usage until such a need is demonstrated.

CIF and MD both seek to minimize BI implementation time and cost. Both benefit greatly from a prototype of decision support interface functionality. The difference between the two in terms of implementation speed and cost involves long-term and short-term trade-offs.

Because of CIF's enterprise scope, the first CIF project will likely require more time and cost than the first MD project, due to increased overhead for making parts of the subject area and business data models as compatible across the enterprise as practically possible. CIF developers should be cautioned against both losing sight of the business unit requirements and trying to perfect the enterprise data model.

In contrast, subsequent CIF projects tend to require less time and cost than subsequent MD projects, especially for business units that utilize existing, robust subject areas. MD developers should be reminded that each subsequent MD project might include nontrivial changes to the already implemented conformed dimensions. Expediting the requirements-gathering and implementation processes may complicate the task of providing consistent and reliable data throughout the BI environment.

The detailed data generally appears once in the CIF (though some denormalization may occur for loading and data delivery performance reasons) and is readily available for any and all data marts, thus minimizing storage space requirements. This nonredundancy precludes storing data (except foreign keys) in multiple places. This feature of the data model also minimizes or may eliminate update or delete anomalies that could occur during cascading processes with redundant data content. These benefits are comprised in the MD architecture.

Volatility

The multidimensional model, especially for the aggregated data marts, is dependent on a determination of the possible questions that might be asked in order to eliminate or reduce the need to reconstruct the fact tables should new or changed dimensions be needed. If a change occurs in a business process (that is, the queries change), then the multidimensional model must be reshuffled or reconstructed. The multidimensional model can certainly be extended to accommodate *some* unexpected new data elements such as new facts (as long as they are at the same level of granularity as the rest of the fact table) and new dimensional attributes. However, at the atomic level, this can be a severe penalty. The fact tables can contain many hundreds of millions or even billions of rows, so a rebuild is not advised. Generally, a new (and mostly redundant) star schema is created when this happens.

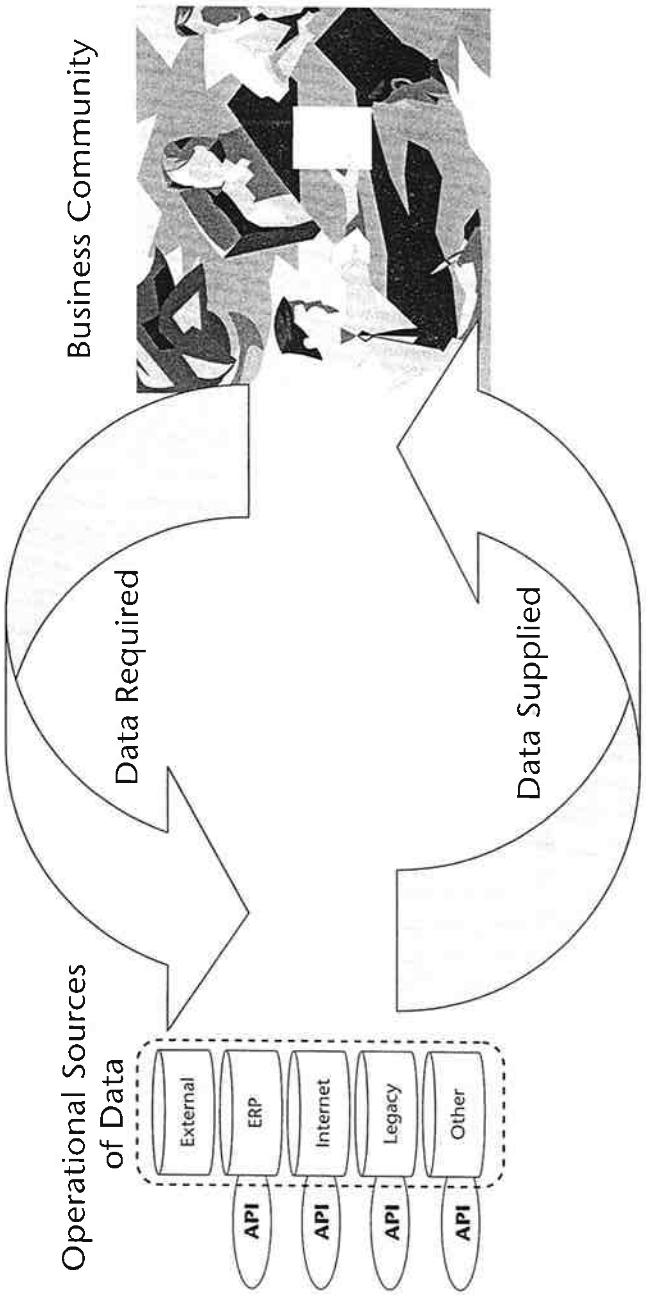


Figure 13.4 Cyclical relationship between business requirements and sources of data.

For the CIF approach, the data warehouse data model is process-free, which removes any biases or hard-coded relationships due to process influences. The data model is dependent on the enterprise's business rules—not what queries will be run against it—for its design. The data model is also far more forgiving of processing changes in the business environment due to a lack of processing bias. Because the model is not designed with any questions in mind, it can supply information for the ultimate data marts through the relatively trivial process of data delivery. If an established data mart requires changes or enhancements, it can be reasonably and quickly rebuilt from the detailed data stored in the data warehouse.

Flexibility

The MD architecture puts a stake in the ground in terms of the design of the entire BI environment. That stake is that all components (except the data-staging area) must be multidimensional in design. This might make sense from an academic standpoint; however, we find in practice that significant and useful technologies can be deployed without this stringent restriction. This is analogous to someone saying that all they have is a hammer and therefore everything must be a nail. If you design your environment using multidimensional designs, then all you will ever do are multidimensional analyses. Nothing more sophisticated or advanced.

The CIF architecture makes no such claim and, in fact, goes to extremes to include the possibility of many different forms of BI analyses. The data warehouse as we have described in this book can support technologies that are not multidimensional in nature. Technologies like memory resident BI tools are certainly not multidimensional. In fact, they require *no* data model whatsoever. Bitmapped indexes and token databases have no need for multidimensional designs. Finally, true statistical analytical tools require flat files or data sets that are not dependent upon multidimensional designs. All are supported with no caveats, biases, or false preconditioning by the CIF data warehouse.

Complexity

Complexities tend to cause fewer problems for CIF than for MD, because the architecture starts with an enterprise-focused, complex data model and then uses it in multiple situations that are usually simpler in design. In the case of creating the multidimensional data marts from the CIF data warehouse, you pull data from a more-complex, multipurpose model into a less-complex one. The data model for the CIF data warehouse minimizes the risk of data inconsistencies because the detailed data in the data warehouse is process-free. In other words, it has not been set up for a specific set of questions, functions, or processes; rather, it is able to supply data for any question or query.

For the MD approach, the multidimensional or star schema data model is easy to understand by the business community. The data model is generally less complex and resembles the way many business community members think about their data—that is, they think in terms of multiple dimensions, for example, "Give me all the sales revenues for each store, in each city and state, by market segment over the last two months." Thus, it is also easier to construct by the IT data modelers. However, given the complexity of an enterprise view of the data as you go from data mart implementation to data mart implementation, retrofitting is significantly harder to accomplish for this architecture. That is why the CIF architecture places the star schema designs in the data marts only—never in the data warehouse itself.

Functionality

The multidimensional architecture provides an ideal environment for relationally oriented multidimensional processing, ensuring good performance for complex "slice and dice," drill-up, -down, and -around queries. All dimensions are equivalent to each other, meaning that all queries within the bounds of the star schema are processed with roughly the same symmetry. We recommend that it be used for the majority of CIF data mart implementations. But do remember that multidimensional modeling does not easily accommodate alternate methods of analysis such as data mining and statistical analysis.

The CIF uses a data model that is based on an ERD methodology that supports the business rules of the enterprise. This type of model is also easily enhanced or appended if need be. Attributes are placed in the data model based on their inherent properties rather than specific application requirements. This is an important differentiator in the BI world because it means that the data warehouse is positioned to support any and all forms of strategic data analyses, not just multidimensional ones. Data mining, statistical analysis, and ad hoc or exploration functionalities are supported as well as the multidimensional ones.

Ongoing Maintenance

There is an old adage: "Pay me now or pay me later." For this final discussion, that adage should be expanded to include: "But it will cost you a lot more if you pay me later." By now, you realize that the whole purpose behind the CIF is to stop the high costs of later constructions, adjustments, retrofits, and suboptimal accommodations to your BI environment. It may cost you a bit more up front, in terms of making the effort to capture an enterprise view of your company's data for your first or second BI implementation. However, BI environments build upon the past iterations and will take years to complete, if it's ever finished. Just as a sound foundation for a house takes forethought and is absolutely necessary for the longevity of the structure, regardless of the changes that occur to it over the years, a well-designed data warehouse data model will serve your enterprise for the long haul. With each iteration, the CIF as your foundation will yield tremendous paybacks in terms of:

- The end-to-end consistency and integration of your entire BI environment
- The ease with which new marts are created
- The enhancement of existing marts
- The maintenance and sustenance of the data warehouse and related data marts
- The overall satisfaction for all your business community members, including those focused on multidimensional analyses

Summary

In this chapter, we described the Multidimensional (MD) and the Corporate Information Factory (CIF) architectures in terms of their approach to the construction of the BI environment. The MD architectural approach subordinates data management to business requirements because its reason for being is to satisfy a business unit within the enterprise. On the other hand, the CIF architectural approach manages data to the subordination of the business requirements because its reason for being is to serve the entire enterprise. The similarities and differences between these two approaches stem from these fundamental differences.

As stated earlier, we find that a combination of the data-modeling techniques found in the two architectural approaches works best—ERD or normalization techniques for the data warehouse and the star schema data model for multidimensional data marts. This is the ultimate goal of the CIF and uses the strengths of one form of data modeling and combines it seamlessly with the strengths of the other. In other words, a CIF with only a data warehouse and no multidimensional marts is fairly useless and a multidimensional data-mart-only environment risks the lack of an enterprise integration and support for other forms of BI analyses. Please develop an understanding of the strengths and weaknesses of your own situation and corporation as a whole to determine how best to design the architectural components of your BI environment. We wish you continued success with your BI endeavors.