CHAPTER

5

Contents

Systems architecture: the preparation for KM

Richard F. Bellaver

5.1 Introduction

This chapter deals with the storage, identification, and quality of data. The emphasis is only on one aspect of the KM benefit spectrum—that of quality. But the emphasis on quality provides the requirements for several of the other topics explained in this book. The chapter also explains the detail data planning, as well as the corporate planning, that must take place for a company to derive any benefit from KM activities.

Besides attempting to optimize on new hardware and software, the phenomenon and the obsession with KM in American industry is having a profound effect on company IT planning. The practicality of distributed data processing (DP) and the desire to take advantage of the latest technology has led many companies to concentrate on cleaning up the databases and restructuring the processing. (Sometimes these things have been done just under the excuse of "reengineering.")

The twofold approach provides a challenge to both corporate IT and business planning communities. The difficulty for planners in attempting to get to KM is an apparent inconsistency in the "quality" definition in IT. Are the advertising descriptions of processing (e.g., new, improved, faster) in conflict with the historical descriptions of data (e.g., identifiable, complete, accurate)?

In those companies undergoing mergers or acquisitions, bringing together diverse IT systems, organizations, and methodologies provides an even more challenging opportunity. Even some of the large, more stable IT organizations have experienced the accordion effect of centralization versus decentralization leading to a similar "clean it up and make it better" situation.

A look at an evolution of the strategies concerning system architecture can be an aid to realizing the problems of getting ready for KM. Computer system architecture is the logical structure and relationship of the data and application functions used to satisfy a company's business requirements. There is a practical architectural technical evolution that can lead to quality-based data, but the nontechnical problems of sharing data for corporate advantage may be more severe than the technical.

Data can be a most valuable asset to a business, and technology can allow shared access to those data faster than ever. If the benefits of data mining are based on shared data, there should be no problem with the methodology being used by most present IT organizations. However, there must be a logical approach to the establishment of data quality procedures before the benefits of mining and warehousing can be attained. At a minimum, interdepartmental battles about ownership of data must be fought, new chargeback algorithms must be accepted, and managers will probably have to learn at least some new coding structures if not some new languages. An examination of the present systems of many companies will establish a base for comparison.

5.1.1 Current architecture

Even with the advent of client-server and unbridled growth in the use of PCs, the current architecture of many large computer systems can generally be defined as mainframe oriented, standalone, and data redundant. This situation did not happen by accident. The early approach to IT for most large companies was to search for projects based on economy of scale. For example, companies looked for large, team-sized applications.

Usually, for order clerks or bill investigators, manual training methods and procedures had been standardized to achieve the last measure of efficiency. Computer mechanization followed the model of these standard operating procedures.

Very large, organizational-oriented systems were built based on the need to increase operational productivity. Generally, the systems used for order processing, dispatching, billing, inventory control, financial control, and many other business functions have been developed using the most efficient technology for both hardware and software.

That technology in the past was basically large mainframes. (These systems are currently referred to as "legacy systems" with somewhat negative connotations. Notwithstanding that they did give us the Y2K problem, my experience is that the developers did the best they could with the ideas and the tools at hand and laid the groundwork for all that has followed.)

In many cases, a total systems approach to mechanization was implemented with that organizational orientation. All data needed to solve a business problem were isolated. The work groups that needed access to the mechanized process were identified, and the rules of the data, the processing, and the communications to solve the specific problem were defined and implemented into one system.

As depicted in Figure 5.1, if customer name was necessary to support system 1, it was acquired, edited, and stored. If these data were needed for system 2, they were also acquired, edited, and stored, but according to a new set of rules (maybe system 1 stored last name first then a comma and first name, while system 2 stored first name and then last name and no comma.) At best, these data were passed off-line from one to N, and then still edited according to system N rules and stored again, usually in a new format.

As a result of the magnitude of the process, the large volume of data, and the limitation of hardware and software capabilities, all aspects of each system were tightly integrated to ensure efficiency of processing time and data storage charges. The cost justification of the project was usually based on increasing human productivity. User departments that paid for the development and the use of the systems reduced cost by reducing human resources. User departments had a very proprietary interest in both the data and the supporting processing system.

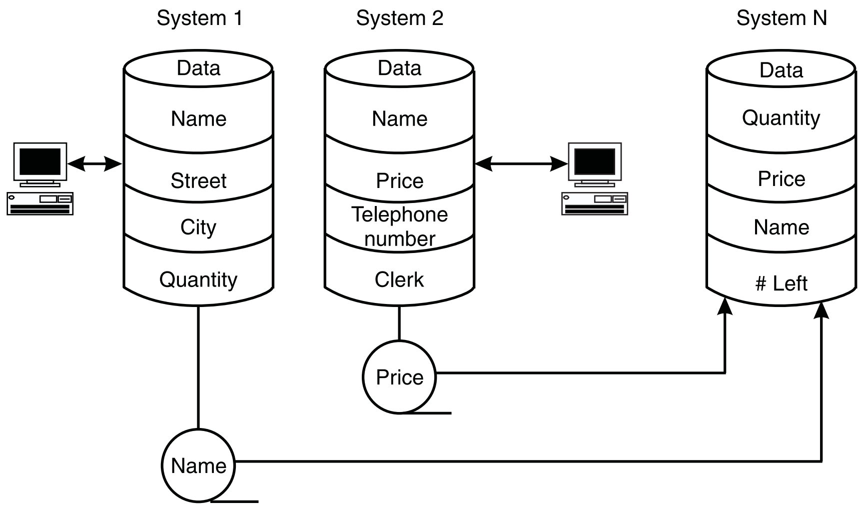

Figure 5.1 Current architecture.

The state of the art, especially the limitations of database management systems and communications software, also left its mark on the current architecture. In many cases, systems developed were efficient, monolithic, inflexible, end-to-end special-purpose procedure speeder-uppers owned by various departments. The computer implementations matched the work, the work matched the organizations, and a degree of stasis was obtained. However, over time, most organizations are subject to significant change.

To contain costs as a corporation moves forward (especially toward centralization or integration), there is a need to increase partnering of organizations, and sharing resources and data is required. Technology cost structure changes and user needs become more sophisticated. Unfortunately to meet this change, most current architectures are characterized by the following;

- Many large standalone systems with individual communications networks;
- Data configured differently for each process;
- Redundant functionality and data;

- Inability of organizations and systems to access other organizations' systems and data;
- A nonquality situation.

5.2 Problems: the opportunities and the challenges

The current architecture picture looks pretty bleak. Must everything be thrown out and started over to clean up and restructure? Economics answers that question. Not many companies have the resources to re-architect their systems from scratch even to take advantage of KM benefits. Cost-effective ways must be found to establish targets for architectural migration. System architecture strategies must provide a transition from the current status to a more flexible architecture that supports organizations and systems working together. These strategies must also maximize the advantages of the following:

- Increasing capabilities of large and small processors;
- Networking capabilities;
- Less complicated programming techniques;
- Understanding the benefits of sharing data (KM);
- Concentration on quality data.

The latter should be emphasized for a simple reason. Data are more permanent than processing. Data are also what tie the corporation together. Some organizations would not talk to each other at all if they did not have to exchange data. Business functionality, although basic, can usually be handled in a variety of ways, but data needed are usually stable. (After all, humans ran the business before computers.)

Improvement of the processing aspects of data processing cannot make up for the lack of historically defined quality of the data. The emphasis of quality for data can be achieved by trapping required data as close to their source as possible and leaving these data in their barest form. The long-range goal of all KM programs must be to have systems designed around the provision of quality data. There are several interim targets that can be used along the way.

An analysis of existing versus long-range target systems architecture yields the following steps for maximizing productivity of existing resources while building target system architectures:

- Search and destroy—eliminate redundancy of functionality and data;
- Surround—add flexibility to present systems;
- Quality data—design, plan, and implement architecture for data quality and long-term flexibility.

5.2.1 Search and destroy—eliminate redundancy

The first architectural strategy is to eliminate functional duplication and redundancy on an application by application basis. Financial and administrative systems are normally a high priority because of the need for common bookkeeping processes, payroll, and personnel systems.

In merged companies, whole systems of the current architecture are subject to elimination. However, usually, under the pressure of time requirements, systems are patched together and new feeder systems are created. Pure duplication in the systems that support operations is usually more difficult to find because the organizations coming together do not quite perform exactly the same tasks.

In some cases, a company is still determining which operations should be merged and which kept separated during IT consolidation. There usually are not a great number of whole systems that can be eliminated or much major functionality disabled, but costs can be reduced by eliminating any duplication of less than whole systems or major functions. However, this is difficult because the current architectures are usually quite inflexible and costly to modify. Therefore, it is usually determined at merge time that the largest part of the current architecture should be continued for some period of time.

This determination seemed quite appropriate at that time. However, company operations start to change. Some of the work done by separate organizational entities starts to be done by consolidated groups. KM work requires that people get data from multiple systems in the current architecture. A strategy has to be developed that can allow users to get to multiple old systems from the same terminal in order to do their work functions.

5.2.2 Defining core data or what is to be mined

The first of the new traumas facing the convert manager is the definition of corporate or "core" data. Core data in its smallest component (in the vernacular of data analysis this is usually third normal form) are those data that are essential to the functionality of the business. They are not financial report data or organizational data or not even previously defined customer data (those data with which the manager is familiar). Those data may not be the previously most used data or most redundantly stored data, although the latter is probably a good indicator. Core data cannot be defined until the company's business functions are well defined.

Defining business functions is difficult and must involve interdepartmental or corporate planners. It's too important to be left to data processing planners. Business functions are distinct from departmental goals or even business objectives. Business functions are the detail of what the business does and how it does it. Hard looks at business functions result in strange new descriptors for the work of the organization, generally under the headings of management control, operations, support, and planning. Only after these overall functions are broken down can planners really determine what data are necessary to perform the functions and determine where to find the source of highest-quality data.

The science of data analysis is becoming better defined each day, but the art of agreement as to which data belong to departments and which are core data is ill defined. The practice of charging back the costs of data processing to the user has reduced IT budgets over the years while bringing in more hardware and software, but it has also engendered a very proprietary feeling toward systems and data on the part of the users. Individual departments believe, since they paid for its conversion, processing, and storage, that they own these data. Upper management intervention is usually required to settle arguments and establish the general rules for the shared use of data. As if upper management intervention isn't traumatic enough, the reason for defining core data is to share these data. This is not always recognized by all participants, leading to psychological manifestations such as sibling rivalry or who has the biggest ego (or budget). Who pays the bill is essential to the establishment of corporate KM policy.

Arguments about the required security of data will also surface and must be resolved. As data are used for more functions of KM, especially competitive advantage, there will be more need for protection of these data from hostile eyes. The problem will be the determination of whose eyes are hostile and what harassment, in the form of password, encryption, or our right to nonaccess will be required.

5.2.3 The data engine

Once it is agreed what should be core or shared, and after the security issues have been resolved, data must go through three steps before storage, as follows:

1. The one source of the smallest component of the data must be identified. Basically this step determines from where and how basic data will enter the storage system. This step is essential in establishing the quality framework of the system. The rules for data integrity must be established here.
2. Standard identification and the catalog structure must be determined. The user must understand what these data really are and how they are identified. The use of "directories" seems to be a way to solve this problem. (See the following section.)
3. A standard output interface must be defined. The calling terminology and the access rules to data must be spelled out fully.

The preceding three steps are usually performed by technicians, but the result generates some syntax that must be understood by managers to get the full benefit of core data. Learning new language and codes has great psychological ramifications.

5.2.4 There must be directories

A directory, or more correctly a metadirectory, is a specialized repository that contains lists of system users and their access rights (for more on directories, see Chapter 8). It also functions as a kind of network white pages, giving users a simple way to locate applications, print services, and other computing resources. Some metadirectories (Microsoft's Windows 2000 Active Directory Services) add IP and network management functions to the directory. The directory also plays a role in system administration, providing IT managers a listing of all the hardware and software assets in a far-flung enterprise. Most important, a directory is a tool to integrate applications and business units that have functioned as standalone systems in the past—a great breakthrough in most organizations.

Today, directories exist in a multitude of applications ranging from a network operating system and asset management system to e-mail and database applications. The cost of implementing and administrating these disparate and often proprietary directories is great. That's why many companies are moving to implement a single, master directory, which can integrate these diverse systems. The business value of a unified directory is compelling: the elimination of redundancy and the automation of business processes across an entire enterprise.

5.2.5 Surround-increase flexibility of present systems

In addition to accessing multiple systems functionality from the same terminal to increase flexibility, KM demands the ability to distribute data from those systems so that users can add their own functionality. The Gartner Group has developed a rather complex seven-stage model depicting the evolution to client-server architecture (Figure 5.2).

Figure 5.2 is a simplified model indicating the shift from a single-purpose data processing system (in many cases that is a current corporate architecture on a mainframe) through a separation of some processing (which may include a server computer or smart terminals) leading to a

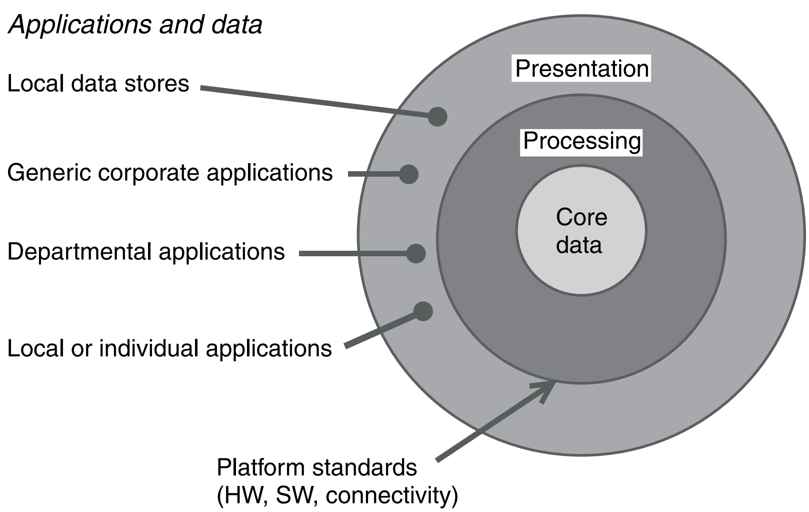

Figure 5.2 The conceptual model.

networked system with the possible use of all the processing tools with access to data that could be stored almost anywhere in the network?

Currently available computer hardware, software, and network products can be used to accomplish a partial distribution of data processing on a step-by-step basis. Legacy systems can be accessed without renetworking every terminal and without multiple log-on and log-off procedures. A server can be programmed to emulate the current terminal/mainframe interface. Data can be accessed using the current data management schema. This can be done with the same or enhanced security requirements with little additional communication time. In addition, with the use of different communication and database management software, file segments or whole files can be downloaded for different local processing.

The surround approach can be implemented with minimal complication to present mainframe processing or database software. The present application programs require modest modifications; some "add-on" programming is required to meet interface standards. Local area networking technology helps resolve communication delays. The distribution of computer and storage devices can provide resources for local development and capability for additional centrally developed systems such as electronic mail and office automation. With the use of a tightly controlled standards process for software distribution and data security, there is potential for departmental reports processing, site-oriented database administration, or other user-generated programming at each site or from a combination of sites. This is the start of KM; however, it does mean additional costs.

The great advantage to the surround approach is that it decreases the need for mainframe program modification. It leaves the current mainframe databases as they are. New user functionality can be recreated using server-based programming, which can be generated faster and cheaper than mainframe programs can be enhanced. By not having to get into mainframe languages or database management systems for every change required by users, analysts and programmers can have more time to apply their knowledge and experience to developing a long-term view of systems architecture.

5.2.6 Quality data structure

The "search" portion of search and destroy takes a detailed look at the processing and data of the current architecture. Surround uses what is learned in search and links data and processing and attempts to meet changed business needs. The long-term view should introduce the concept of a functional orientation as opposed to the traditional organizational approach to doing business (that's what reengineering is all about).

The theory is to examine what functions are required to best serve the needs of the corporation's customers and then to determine how best to accomplish those functions. A functional model of the corporation should be constructed. When the functional model is understood, data needed to support the business functions must be defined and the source described. These data must then be standardized and a corporate data directory built to ensure that these data are of the highest quality and that these data stay that way.

As discussed by Richard Y. Wang in *Communications of the ACM*, it might be necessary for corporations to manage data as they manage products. Many companies, thanks to the Japanese and Dr. Deming, are taking a total quality management approach to more of the business. This means total work force commitment, management responsibility, and a complete customer focus in attacking work processes. This kind of approach is necessary in defining core data and even more so in building the systems and procedures to keep quality data.

5.2.7 Separate the data from the processing

Close examination of data in most current system architectures indicates several potential barriers to data quality. The search to eliminate functional redundancy usually identifies significant data redundancy or apparent redundancy. There are multiple systems storing the same data (Figure 5.1) but coming from different sources. There are multiple systems storing different data from the same source. There are systems deriving and storing summarized data from detail for one business purpose and other systems deriving and storing a somewhat different summarization to be used by a different business function.

Although data editing and quality checking were stressed when individual systems were built, the combination of data that may be used for KM purposes was not preplanned or coordinated. An obvious problem with the current architecture is the cost of resources for processing and storage for the redundant data. The more serious problem, however, is the lack of confidence generated when a user confronts nonmatching data while trying to solve a customer problem. Redundant data or apparent redundancy are not quality data.

Use of poor quality data causes slow reaction to customer needs and poor customer satisfaction. Resolution of the data redundancy/quality problem is simple—separate data from the processing and build a data engine, as mentioned previously.

5.2.8 Conceptual model

Thinking about data separated from processing leads to a layered approach. This approach is only feasible through well-defined, strictly enforced standards dealing with hardware, software, and connectivity. These rules form a standard operating environment, which must be in place to allow access to shared data. A conceptual systems model depicts three layers, as follows:

1. The core data necessary to accomplish business functions;
2. Processing of transactions necessary to get core data into and out of databases;
3. Presentation or other manipulation of core or local data required by the user (Figure 5.3).

5.2.9 Supporting technology

The conceptual model does not imply any specific hardware implementation, but certain inferences can be derived based on changing technology. In the current architecture, terminals connected to mainframes are used to gather, edit, and store data (Figure 5.3). Mainframe programming formats reports and rearranges data on terminal screens. Mainframes summarize and restore data. All data processing and presentation can be done with mainframe programming. With the capabilities of new technology, opportunities are available to use mainframes, servers, and personal computers (PCs) to greater advantage through networking.

To store core data in the smallest logical component, find these data, and provide all the required derivations, it will be necessary to use complex relational data structures and directories. The processing power required (even with special database machines) indicates that mainframes may be required to build and maintain large shared databases. However, the processing of those data, or the manipulation of the transactions that get these data into and out of the system, could be done with servers.

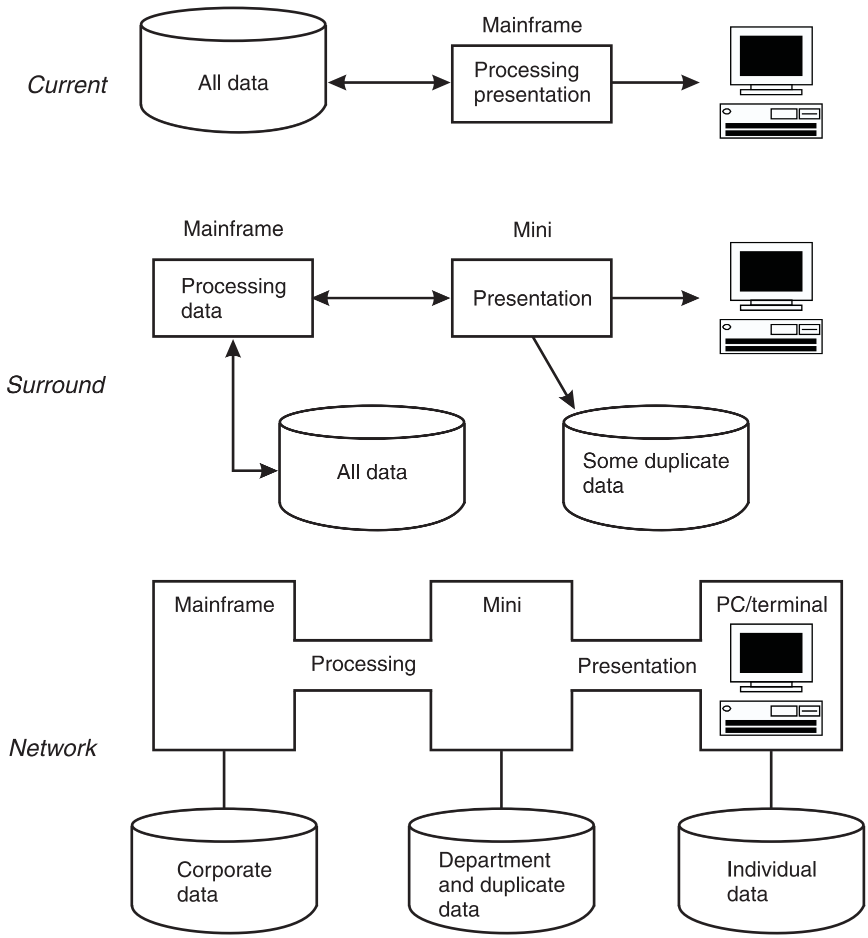

Figure 5.3 Technological architecture.

5.2.10 The "look" or the "content"?

Once consistent quality data are available, presentation—the way data look on output—can be driven closer to the user to provide flexibility. All of the formatting of these data can be done outside the mainframe, thereby reducing the load on the communications facility. (Terminal emulation mode requires that all screen format characters, as well as data, be exchanged with the mainframe. Studies indicate that by only sending data, communications requirements can be cut by orders of magnitude.) The programming could be done in a server for a group of users to analyze data in the same manner or in a networked PC if the display is for an individual.

Although this new technological approach (the processing definition of quality) is important to architectural planning, it is more important to analyze functions and data required for functions before jumping into the technology. The surround approach and the uses of new technology will produce some better efficiencies and add some flexibility at a reasonable cost (if local enhancement capabilities are very carefully accounted for), but the quality roadblock cannot be eliminated unless corporate data are standardized and made available for all processing. In the long run, a redesign of major systems around the use of quality data is required. A combination of moving to new technology while achieving quality data is ideal.

5.3 Implementing a KM strategy

The idea of redesigning major systems around quality data or anything else seems to be an anathema in these days of cutbacks. A greater problem is that data planning is difficult on a corporate scope. The whole is too big even for the best corporate and IT planners. However, planning done on an organizational basis will bring about another generation of new, improved, and faster nonquality systems. It is possible to identify clusters of data to single source and start sharing. Search will identify data that are currently being shared. Savings achieved in elimination of redundancy in destroy process can be used to pay for extra hardware needed for the surround process.

Each of these strategies refines the data sharing process until it becomes practical (either through cost justification or some less tangible value judgment) to separate certain specific data and build a data engine. It is impractical to reimplement most operational support systems at one time to make a great leap forward. The better plan is to move from the current architecture to a series of interim architectures, each becoming more quality-data oriented (Figure 5.4).

Search and destroy should be pursued not only to save money but also to identify and to start the elimination of redundant data. A logical separation can begin with implementation of the surround approach for specific functions. Most of this hardware can remain in place to transfer

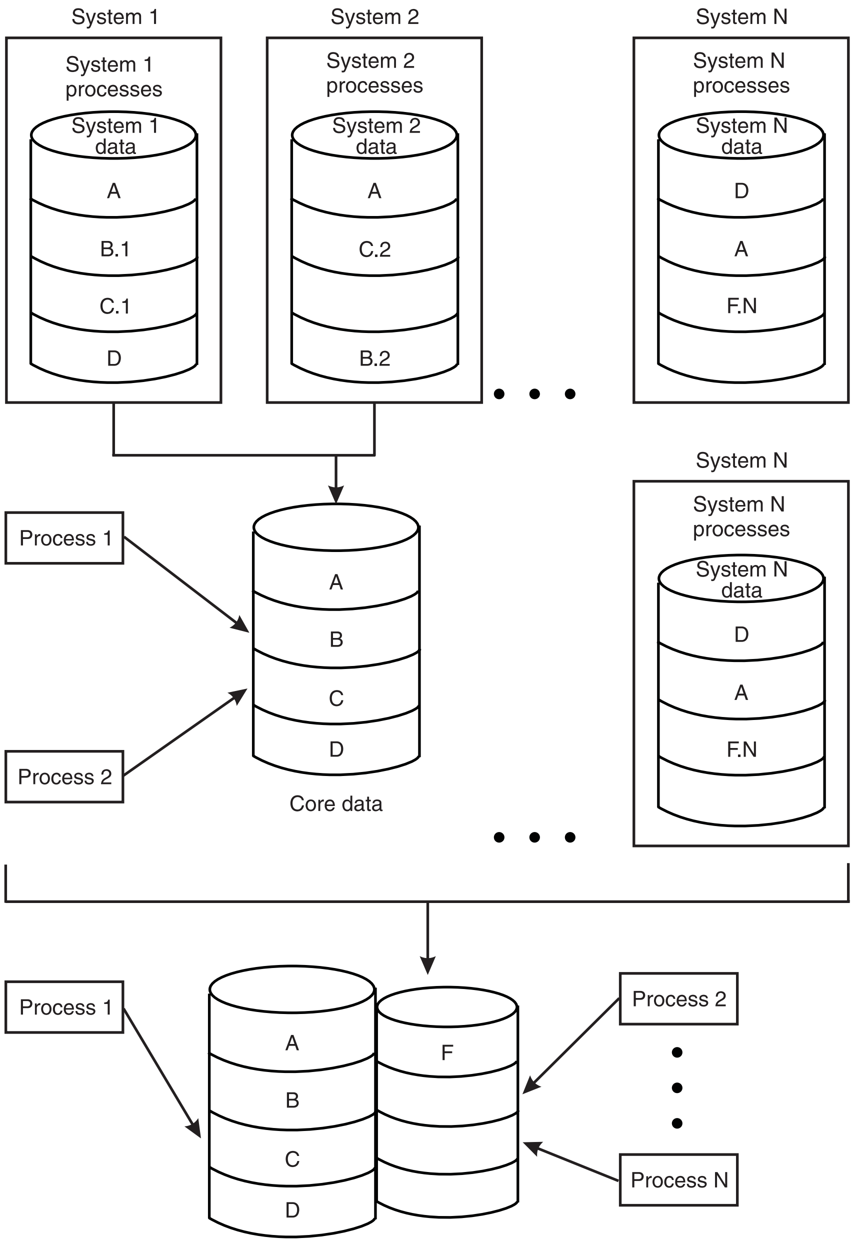

Figure 5.4 Architectural revolution.

to processors in the conceptual model. Concentration on quality data can begin by examining each new mechanization project in light of its use of standard core data.

As practical clusters of data are identified, data engines should be designed that provide the storage structure and distribution of corporate data using the presently installed mainframes. All present database systems should be examined to determine modifications needed for interim data systems and methods for converting and merging data into the next generation of engines. Over time, with new systems and high-priority modification required by present systems, the goal of quality data can be reached.

All aspects of quality are important to IT, but data quality is essential. Current systems architectures do not support economical data sharing or take advantage of new technology. Future systems will be designed around the use of quality data stored in their smallest component and available for all processing. Networking will provide the advantages of the best properties of mainframes, minicomputers, and PCs. Surround structures are an interim approach, providing continuing use of the current architecture while laying the hardware base for the transition to transaction and presentation processing of the future. A plan of migration can be developed targeting an ever increasing sharing of data until the future design can be realized.

5.3.1 KM side benefit

The technical architecture of the evolved storage structure shows the repository of core data, but it also implies that core data can be duplicated and that departmental and individual data can be defined. This means that after definition of core data there must be further definitions of the storage structure. Core data will be used for traditional corporate data processing—that is, turning these data into information for corporate reports, payroll, and so on—but data in its information (output or presentation format) will never be kept in a corporate information base. A dollar-saving aspect of getting ready for KM is that only data will be kept.

Summarizations will never be retained in the corporate database. They will just be generated on output. Customer bills will never be found in bill format in the corporate database. Storage backup will be less data intensive and cost less, but users must now address that database with the proper language to retrieve year to date or quarterly results or to create a bill status when handling customer inquiries. A hard copy of a customer bill will never be retained. Archiving, if the medium is cheap enough (microform or laser storage technology), can be on an information basis.

"What if" games will now be done outside the mainframe on departmental processors (if data must be shared) or all the way out on individual PCs. Departmental data not needed for corporate processing (i.e., team sales results, training information, bowling scores, and so on), can reside only at the departmental level. Corporate data can be duplicated and reside at the departmental level if aging information is applied by the departmental processors when these data are downloaded. (Warnings that these data were downloaded at a certain time or that data are only valid until a certain time must be applied.)

Another benefit of KM will be found as corporations assess knowledge communities (as mentioned in Chapter 12). This benefit will be moderated by the new storage requirements demanded by employees trying to keep text and multimedia records downloaded from the Internet. More sophisticated identification schemes will be needed to share these data across the organization.

5.3.2 Data quality rules

Here's one last thing about data quality. There seems to be a relationship between quality data and their use. Over time data seem to become less valuable to the corporation. Out-of-date information certainly is of little value except for historical purposes. But even user data, if not updated or refreshed, also tend to cause problems. If this is true, then, just as with one's desktop or attic, some purge or "spring cleaning" should be scheduled. Ken Orr, in *Communications of the ACM,* has addressed this problem in his list of six rules for data quality. These rules are as follows:

1. Unused data cannot remain correct for very long.
2. Data quality in an information system is a function of its use, not its collection.
3. Data quality will, ultimately, be no better than its most stringent use.
4. Data quality problems tend to become worse as the system ages.

5. The less likely some data attribute (element) is to change, the more traumatic it will be when it finally does change.
6. The laws of data quality apply equally to data and metadata (directories).

5.4 Conclusion

There are many problems with rearchitecting for KM. The changes will be significant, (read: hard to explain and hard to sell). Individual or private data and individual programs will be stored only in PCs with the tightest security precautions to avoid upward contamination. Mail and/or announcement services will originate at the PC and can still be hubbed on mainframes depending on the communications networking scheme, but any storage on mainframes will be physically separated from corporate data. Formatted data, such as expense vouchers, will be input from the PC, but only corporate required data will be passed up the line. Other data will be kept in departmental files or even locally. If it is appropriate to disburse from the departmental level, the check creation or an electronic funds transfer can take place there (hopefully speeding up the process) with only financial data going to corporate. Decentralization with centralized quality data will gain a new meaning.

The psychology of change impact of such a data and information processing hierarchy as described is traumatic. Besides the knowledge of the standard identification of data needed (new codes and language) for sharing, an understanding of the storage hierarchy must be in place. Individual managers must understand which data will be shared and where to find these data. They must also understand how to legitimately avoid the sharing of data. Auditors, as the managers, will no longer be able to pull hard copies of vouchers from departmental files. They must know how to navigate to the furthest ends of the storage structure to get all the comments added by originators. *Harvard Business Review* of March/April 1999 has a good article on the strategy of this process.

Storage pricing structures will vary greatly depending on the location of data and their utilization. New charge-back allocations will have to be developed. All this takes place after the trauma of core data definition, with all its internecine battles and, yes, even the need for top management involvement in really determining the functions of the business, has been faced.

The solutions to these problems, really introducing the concepts of KM into an organization, create a new set of promises of quality, strategic impact, and even cost savings available to potential users. In my 40 years in the business world I have seen the leaders of IT, in many companies, go from radicals demanding change to conservatives embracing and fostering the status quo. It is time for that leadership and even more important general corporate leadership to get radical again. The need to prepare users for the psychological aspects of the great opportunities has never been so important. Companies must be prepared to face the challenges and the time required to do all the interdepartmental planning work required to gain the potential advantages. These factors must be understood by all before it is appropriate to move ahead.

Selected bibliography

Publications

KMWorld, November 1998

Communications News, December 1998

Data Quality for the Information Age, Artech House, 1998

Communications of the ACM, February 1998

InformationWeek, December, 1998

Beyond Computing, May 1998, March 1999

CIO, February 1999

Network Computing, January/February 1999

Harvard Business Review, March/April 1999

Enterprise Systems Journal, May 1999

Internet Web sites

www.Datawarehousing.com

www.KMWorld.com

www.CIO.com/forums/data

www.Pmp.starnetic.com/larryg/index

www.DW-institute.com

www.Datawarehousing.org

www.Dbnet.ece.ntua.gr

www.Altaplan.com/olap