JUNE 1986 Vol. 9 No. 2

a quarterly bulletin of the IEEE computer society technical committee on

Database Engineering

Contents

SPECIAL ISSUE FROM THE FOURTH DATABASE DIRECTIONS WORKSHOP: INFORMATION RESOURCE MANAGEMENT—MAKING IT WORK

Database Engineering Bulletin is a quarterly publication of the IEEE Computer Society Technical Committee on Database Engineering. Its scope of interest includes: data structures and models, access strategies, access control techniques, database architecture, database machines, intelligent front ends, mass storage for very large databases, distributed database systems and techniques, database software design and implementation, database utilities, database security and related areas.

Contribution to the Bulletin is hereby solicited. News items, letters, technical papers, book reviews, meeting previews, summaries, case studies, etc., should be sent to the Editor. All letters to the Editor will be considered for publication unless accompanied by a request to the contrary. Technical papers are unrefereed.

Opinions expressed in contributions are those of the individual author rather than the official position of the TC on Database Engineering, the IEEE Computer Society, or organizations with which the author may be affiliated.

Membership in the Database Engineering Technical Committee is open to individuals who demonstrate willingness to actively participate in the various activities of the TC. A member of the IEEE Computer Society may join the TC as a full member. A non-member of the Computer Society may join as a participating member, with approval from at least one officer of the TC. Both full members and participating members of the TC are entitled to receive the quarterly bulletin of the TC free of charge, until further notice.

Dear TC DBE Member or Correspondent:

I attended the meeting of the Technical Activities Board of the IEEE Computer Society at COMPCON on March 5th, 1986. Important changes announced in regard to our Technical Committee are a significant (30%) reduction in Computer Society contribution to the Technical Committees. Our Database Engineering Bulletin is a costly item (estimate $18,000 per year), whereas a fair and balanced support formula would only net us about $1,200. The formula proposed is based on a fixed amount per TC plus a fixed amount per TC member who is also a Computer Society member. It appears we have about 500 Computer Society members and another 1000 correspondents who receive the bulletin. The numbers are not up-to-date because no new members have been added to the files for about 3 months due to a staff shortage at the IEEE Washington office.

There are other sources of income if we want to continue this bulletin and its quality.

1. Charge recipients of the DBE Bulletin.
   a. Charge all recipients, but provide an additional free newsletter to IEEE CS members.
   b. Charge a differential amount to members.
   c. Charge non-members only. This latter would require some success with points 2 and 3.
2. Operate conferences, tutorials, and workshops to yield a surplus. The prime candidate is the Data Engineering Conference. It is currently not budgeted to yield the required surplus.
3. Negotiate with a commercial publisher distribution of the DBE Bulletin in a way so that we reduce the publishing and distribution cost.

Charging must recover most production costs and the costs of billing (about $5). A possible amount would be equivalent to typical ACM SIGMOD membership fees, i.e., about $15–$20. We realize that this would be a heavy burden, especially on our foreign correspondents. Note that the ACM SIGMOD membership fee was only $3, but is now increasing to $15, and that the ACM SIGMOD Record just started publishing again.

The new environment requires additional effort to run the TC. My own term is past, but we are happy that Sushil Jajodia has agreed to accept the chairmanship. I will, of course, remain available to help in the transition. I expect he will look for help and appoint a vice chairman and perhaps a treasurer and secretary.

A plan to share both a joint TC and ACM SIGMOD board member are still in abeyance, although a number of you volunteered. The ACM and IEEE have decided to initially only have one group interact strongly, not our group.

Please feel free to correspond with Sushil Jajodia (Computer Science and Systems Branch, Code 7590, Naval Research Laboratory, Washington, DC 20375-5000) or with John Musa, Vice President for Technical Activities of the IEEE CS (Bell Laboratories – 3A332, Whippany Road, Whippany, NJ 07981) about the issues facing the TC, and feel especially free to volunteer your services to help the TC on Database Engineering maintain its stature. We have, I think, the best bulletin, we are the major sponsor for what is becoming an excellent conference, and are co-sponsoring many exciting activities.

Gio Wiederhold

Letter from the Guest Editor

This issue of *Database Engineering* is devoted to the topic of Information Resource Management (IRM). As pointed out by Bob Curtice in the introduction, the goals of information resource management include:

- Managing the information resources for an entire organization. Various forms of information -- computerized information bases, non-machine processable information, policy guidelines and documents, verbal and written communications -- are all within the scope of IRM.
- Providing the best access to the spectrum of users while maintaining integrity and security of the information.
- Being consistent with and sensitive to the business needs of the organization.

As we consider the development of new technologies to design better databases, better systems to manage them, and better facilities for access and control, it is essential that we do not lose sight of the forest for the trees. Since a majority of the readers of *Database Engineering* are working toward one or more of the above subgoals, we felt it was worthwhile to put before them a macro-viewpoint on database management as a subset of information resource management. Our hope is that this will help in perceiving and evaluating the technical issues in a broader perspective and with an enhanced pragmatism.

The current issue is an edited version of the deliberations at a three-day workshop on "Information Resource Management--Making It Work" held at Ft. Lauderdale, Florida, on October 21-23, 1985.

This was the fourth in a series of workshops sponsored by NBS. Titles and dates of the previous Data Base Directions (DBD) Workshops are:

- *Database Directions: The Next Steps*, DBD-1, October 1975, (Published as *ACM SIGMOD RECORD*, 8, 4, Nov. 1976).
- *Database Directions: The Conversion Problem*, DBD-2, November 1977, (Published as NBS Special Publication, 500-64)
- *Data Base Directions, Information Resource Management—Strategies and Tools*, DBD-3, October 1980 (Published as NBS Special Publication, 500-92).

The workshop was attended by about 70 invited participants. Bob Curtice, Elizabeth Fong and Alan Goldfine were the chief organizers; I acted as the IEEE liaison. Four working groups were formed and group leaders were chosen about six months before the workshop. The group leaders drew from a pool of suggested names as well as invited persons of their own choice. We were able to get a very good representative group that included DP management and technical management from industry and government, academicians, consultants, and some not-so-easy-to-classify people. The four groups and their leaders were:

- IRM in the 1990's: Dan Appleton
- IRM and the System Life-Cycle: Beverly Kahn and Sal March
- Technologies for IRM: Alfonso Cardenas
- IRM in a Decentralized and Distributed Environment: Olin Bray.

Each of the working groups held discussions independently for one and a half days in different formats. The working groups summarized their discussions by making presentations to the entire body of participants on the last day of the workshop.

After the workshop, the working group leaders put together reports of their own groups using contributions from their members. These were edited by Fong and Goldfine. Some further editorial changes suggested by me have been incorporated.

What we are able to present to you is a version of the "position papers" of each of the working groups. Changes made have been mostly editorial; the original content represents a collective set of opinions that emerged from the groups. The frank and intense discussions have brought many open problems to the fore, with some pointers to solutions.

I appreciate the hard work put into this workshop by the organizers and thank them for the opportunity to share it with our readers. We hope the readers will find this issue informative, interesting, and thought provoking.

Sham Navathe
Guest Editor

May 1986
University of Florida
Gainesville, Florida

DATABASE DIRECTIONS
INFORMATION RESOURCE MANAGEMENT—MAKING IT WORK

Elizabeth N. Fong,
Alan H. Goldfine, Editors

Institute for Computer Sciences and Technology
National Bureau of Standards

ABSTRACT

This report constitutes the results of a three-day workshop on how to make information resource management work, held in Fort Lauderdale, Florida on October 21-23, 1985. The workshop was sponsored by the Institute for Computer Sciences and Technology (ICST) of the National Bureau of Standards (NBS), in cooperation with the Association for Computing Machinery, the IEEE Computer Society, and the Federal Data Management Users Group.

Patterned after the three previous Data Base Directions workshops, this workshop, Data Base Directions: Information Resource Management—Making it Work, evaluated current practice to identify problem areas, reviewed important technologies and tools and when to apply them to information resource management, and explored the motivation and inhibitors to decentralized and distributed environments. The approximately seventy workshop participants were organized into four working panels, which met to discuss IRM in the 1990s, IRM and the System Life Cycle, Technologies for IRM, and IRM in a Decentralized and Distributed Environment.

1. INTRODUCTION

Robert M. Curtice

Biographical Sketch

Bob Curtice has, for 20 years, been a consultant with the firm of Arthur D. Little Inc., where he specializes in technical and management issues of information resource management. He has assisted scores of client organizations in the adoption of data management systems, establishment of data administration and database administration functions, and the adoption of the systems life cycle for IRM.

Mr. Curtice is co-author of Logical Database Design, (Van Nostrand Reinhold, 1982) a book that explains a unique approach to logical data modeling. His next book, entitled Strategic Value Analysis - A Modern Approach to System and Data Planning, will be available from Prentice-Hall in the Spring of 1986.

Mr. Curtice holds a B.A. in Mathematics and an M.S. in Information Science, both from Lehigh University.

The rapidly changing nature of information technology tends to inflict schisms of various kinds upon our profession. Before graduates of our universities and technical schools can practice with skill and confidence what they have learned, new methods and techniques have evolved and are being taught to the next group of students. Conversely, manufacturers and software vendors continue to make yesterday's products (which we have barely begun to master) obsolete. The academician, the vendor, and the practitioner are at different places; even within the practicing community, levels of experience, understanding, tools, methods, and strategies abound. Admittedly, we are forced into a somewhat haphazard approach to plying our trade. Nevertheless, we can and should do a better job of exchanging ideas and learning from each others experiences.

Editors' Address: National Bureau of Standards, Gaithersburg, MD 20899, (301)921-3491.

The National Bureau of Standards has sponsored four Data Base Directions Workshops over the past 10 years. These meetings offer one of the few opportunities for members of the academic, commercial, government, and vendor communities to come together and share ideas and experiences. This workshop, the fourth in the series, focused on the issues of Information Resource Management—Making it Work.

The goals of information resource management include:

- Managing information independently of organization and application
- Defining and structuring information to meet real business needs
- Enabling end-users to access their data directly, when so authorized
- Ensuring the security and integrity of information on an enterprise-wide, consistent basis.

These goals have been articulated for a number of years and are widely accepted. Yet, we are no nearer to achieving them in most organizations than we were five years ago. Why? This question is the main theme of the Fourth Data Base Directions Workshop. It is not to define the goals of IRM nor to explore why it is desirable, but to examine where we are realistically and what is needed to move ahead—in other words, how do we make it work?

I am impressed by the sincere professional interest in the subject matter at hand taken by the many participants in the Workshop, and with the ideas, thoughts, and written material they generated in a few days. I am convinced that the confluence of so many interested and capable people sparked ideas. I for one came away with a renewed appreciation of the high quality of all the participants, and benefited from the frank and intense interchange of ideas. I am sure others did as well. We owe thanks to the National Bureau of Standards and its staff who make the experience possible.

2. KEYNOTE ADDRESS

Eugene Bloch

Biographical Sketch

Eugene Bloch is the Director of Corporate Information Systems and Services for Allied Signal, Inc., a corporation that has grown through acquisitions over the past six years by six fold. He is responsible for Corporate-wide long range planning and control of the information systems function. He joined the company's Chemical Sector in 1969 as an operations research analyst. He has held his present position since 1979.

Previously, Dr. Bloch was with General Dynamics Corporation as a control systems engineer.

Dr. Bloch is a graduate of the Massachusetts Institute of Technology, where he received his MSEE degree in 1958. He is also a graduate of NYU's Courant Institute of Mathematical Sciences where he received a Ph.D in 1969.

2.1 A PLEA

I am pleased to be here today to address such a prestigious and talented group at the outset of this important conference.

The role of a keynote speech is usually to provide a "beacon" that illuminates the key issues to be addressed, and to set the stage for the deliberations that will follow. Unfortunately, the company where I work is not one of the handful of companies who have realized the promise of Information Resource Management, so that I can't light your way. However, as a representative MIS manager who comes from the real, and sometimes dark, world of systems development, operations, budgets, demanding users, and application backlogs, I can report that our current methods are generally inadequate and deliver a one word message to let this conference know that what you are trying to accomplish—"making it work"—is critically important. That message is "HELP"!

Beyond that plea, I'd offer a perspective, based on experiences and observations, of what I perceive to be some barriers to making IRM work—none of them will be a surprise to this group but they are perhaps more significant and difficult to maneuver around than one might at first suspect.

2.2 BACKGROUND

First, let me tell you something about Allied-Signal. We are, today, a $16 billion diversified corporation that operates in four business sectors: Aerospace, Automotive, Chemicals, and Industrial & Technology. You may have never heard of us—but are perhaps familiar with some of our businesses: Allied Chemical (formerly Allied Chemical Corporation), Bendix, Fram, Ampex, Garrett, Fisher Scientific and others. For planning purposes we view the corporation as comprised of approximately 75 entities or Strategic Business Units (SBUs). Although we are by no means a holding company, operating responsibility is generally pushed down to the SBU level. We have over 40 major data centers worldwide. They are managed in a decentralized manner. Day-to-day operations, systems support, and development is done locally and the MIS managers report locally to divisional or sector management.

I manage a small staff of consultants/planners in MIS and telecommunications in the Corporate office. We are responsible for long range planning, matters of policy and control, and review and approval of major DP projects. We believe that the closer the MIS function is placed to the users—ideally the SBUs—the more effective the function will be, even though we may spend more money than if a more centralized strategy were employed.

When we speak of an SBU, we mean a business entity in a definable market within some industry, so you can talk about sales, distribution, production, engineering, and staff support functions.

2.3 BARRIERS

Clearly, the domain to apply IRM in our corporation is at the SBU level. We have attempted this in 3 business units. I will describe these experiences a little later, after several general comments about barriers.

A major barrier to making IRM work is the credibility of MIS to be the change agent for IRM within the enterprise, as logical as that might seem to be. What is the profile of the typical MIS organization that would be prone to this problem? They have a technology, not a business orientation. They talk in terms of operating systems, CICS, DBMS, COBOL, and not business. They tend to be focused toward finance and accounting applications, partly because of the history of their evolution within the company—in fact they probably report to the Controller. There is nothing wrong with such a reporting relationship, unless it turns out that, for example, a sales person can't get a critically important report because the MIS staff is putting the general ledger system on-line. It is a matter of priorities. They are seen to be unresponsive to demands for needed information—things take too long to get done. They resist change—they are doing things the old way.

This crisis of credibility can be applied to the data processing industry as a whole. Consider the applications for office automation. Our studies, and those of many others, show that the most important need for office workers, after personal computer applications, is for access to information. That is no surprise to this group, but there are two surprises for the unsuspecting user. First, he doesn't get access to information the way he wants it because it's not organized properly (it's not in a database or maybe it's in too many databases). Second, he may instead get applications such as electronic mail and "calendaring" which may be "nice to have" but really are of secondary benefit, compared to his critical needs.

The user is like a person in the middle of a lake drowning. Standing by on shore is the MIS manager, saying, in effect, "I don't have a life preserver to help you out, but if you make it back to shore I've got a nice martini waiting for you." It's a matter of priorities.

This kind of hype is not new to business—there have been many unfulfilled promises in the past, including the MIS dream of the 1960s when we read about top executives running the factories from terminals at their desks with armies of middle management people eliminated.

Now, it's important not to overstate the case but one should be sensitive to the possibility that if MIS tries to "sell" IRM to management they must be prepared to effectively deal with the issue of credibility. Will an unsympathetic management perceive IRM as just another panacea? If so, MIS should find a business oriented champion instead.

A second and related barrier is an organizational environment which may not be ready for IRM. The notion that data is a resource, just like equipment, people and money—and therefore must be managed—is one that most business managers would support. However, how many companies behave as if they support it?

MIS is too often managed as an overhead function, with tight controls and cost containment on a year to year basis, without a strategic view. There is often not a commitment to planning for the business. Even if there is such a commitment, the idea of building an information systems plan to support that business plan may be perceived by the business people to be unnecessary or irrelevant. So, it is possible that the culture within the enterprise may not be ready to accept the concept of IRM. This may also be the situation within MIS if there exists an inflexible adherence to traditional methods of systems development, lack of use of modern tools, an excessive control orientation in management style, the absence of database orientation, and an organizational structure that separates the jobs of programmers and systems analysts. What characterizes a proper environment for IRM, in my opinion, are computer oriented business people and business oriented computer people.

A third barrier is our ability to absorb advances in technology. Rapid change in computer technology can wreak havoc on information systems plans. For example, who in 1980 included PCs in their five year plan? It could happen that the economics of a centralized on-line system are totally destroyed by using a distributed approach via PCs.

In theory, an IRM plan should transcend issues related to the development and direction of computer technology, but in practice, this may not be a valid assumption. In addition, such issues obscure the business focus which is required to be successful with IRM. A further problem is that the new technology may not work if not applied properly. For example, the so-called 4GLs have solid potential for productivity gains but there is a downside; you may have heard about the problems the Division of Motor Vehicles in New Jersey has had recently in its use of an on-line system written with a 4GL.

Finally, a natural barrier to making IRM work is the tremendous investment in current systems. In most cases, the cost to replace this investment requires that the approach be evolutionary. We can't just start all over, but it is very difficult to proceed by evolution rather than revolution.

Another aspect to this issue is the dilemma between an IRM based plan and the use of purchased software packages—that is, if an organization builds a data model and is ready to build the applications, how, if at all, do purchased products fit into the structure?

This dilemma is also a credibility issue; over time, companies have come to accept the idea, often with the strong support of MIS, that purchased software is an economic alternative to custom systems built in-house. Are the MIS people now changing their minds on this?

2.4 EXPERIENCES

These barriers to making it work, credibility, environment, technology, and investment are very real to us at Allied-Signal since we have encountered them as we have tried IRM planning at several SBUs. At one unit that produces complex instrumentation, an enterprise-wide blueprint for data was developed. It was an intensive process that took about 10 months, with the assistance of competent outside consultants. The project had received high level management endorsement and some "seed money" from upper management to get it going. However, the results are not really being used. Why?

MIS supports this SBU and two others that are located together geographically. Although an MIS analyst had been assigned to work with the project team in the SBU, the environment within the MIS unit needed to be changed and it wasn't—they weren't involved in the planning process. Second, the MIS unit elected to buy a packaged manufacturing system based on combined needs of the three SBUs and simply abandoned the IRM blueprint. Finally, since the seed money was provided from outside the SBU, there was little commitment from within to making it work. There have certainly been benefits from the project; it helped to set and justify some priorities and it gave the SBU people an appreciation for the structure of their data, but the effort fell short of original expectations.

Another enterprise in the Allied-Signal family is in the distribution business. They have a home grown early 70s vintage order processing system that is the heart of their business—allowing them to achieve reasonable margins on very small orders. It includes the ability to allow access from terminals at the customer site. The system is old and inflexible and needs to be upgraded—a prime candidate for IRM planning. However, MIS can't sell it; their problem is credibility; the SBU finished a "conventional" planning approach and management is now ready for results. A database management system has been evaluated and purchased and there is no patience for more studies.

A third unit that produces electronic components for the defense industry has constructed an enterprise wide blueprint of their data and is moving ahead into the build phase. This unit appears to have none of the problems the other two had. It is now more a question of taking the project forward.

2.5 SUMMARY

The barriers that I have described, real or perceived, must be removed if the promise of IRM is to be fulfilled. I think that the working panels of this conference have posed the right questions and I challenge you to develop answers and approaches to continue the positive momentum achieved in the previous Data Base Directions conferences. We need your help in making IRM work. I look forward to the progress you will make in the next few days toward that goal.

3. IRM IN THE 1990s

Daniel S. Appleton

CHAIRMAN

Biographical Sketch

Daniel S. Appleton is President of D. Appleton Company, Inc. (DACOM). He specializes in industrial modernization and data resource management. Prior to establishing DACOM in 1979, Mr. Appleton was Director for Strategic Business Planning at the Borg-Warner Energy Equipment Group and Director of Management Information Systems for the eight worldwide manufacturing facilities of Byron Jackson Pump. He has also been Manager of Systems Development at Litton Ship Systems, and has worked for the CIA and in the office of the Assistant Secretary of Defense, Comptroller. Mr. Appleton received his B.A. degree from the University of California, Berkeley, and his M.B.A. from American University in Washington, D.C. He is a Fellow of the Institute for the Advancement of Engineering, the Chairman of the Technical Council for the Computer and Automated Systems Association (CASA) of the Society of the Manufacturing Engineers (SME), and an active member of both the IEEE and the American Association for Artificial Intelligence. Mr. Appleton has published numerous technical papers and articles, and he is the most published author in Datamation Magazine, having had 19 articles published.

PARTICIPANTS

Ray Bell	Jerry M. McClure, Jr.
John Berg	Nancy McDonald
Bob Benson	Alan G. Merten
Thomas P. Cahill	Gary Mortenson
Jack Clowdus	Marilyn Parker
John W. Coyle	Edward W. Patneaude
Bob Curtice	Warren B. Rigdon
Tor Guimares	Robert S. Rosenzweig
Jeffrey A. Hartigan	Thomas L. Ross
Raymond J. Hollenback	John Zachman

3.1 INTRODUCTION

The charter of this working panel was to determine the economic, political, and technical trends that would shape the IRM function and organization over the next decade. The panel consisted of 20 professionals—11 practitioners, 3 consultants, 3 academics, and 3 vendors.

The panel perceived IRM as a cultural issue. In order to examine this issue, the panel accepted a conceptual model that defined three levels of culture (see Figure 3.1).

The highest level of culture is the value system. The value system defines what is right and what is wrong in the culture. It is a set of principles and philosophies.

The value system drives the process structure of the culture. The process structure contains all of the cultural institutions, including its organization, its planning system, its control system, and its administration system.

```
 _____________
 |  VALUES   |
 -------------
       |
       v
 _______________
 |  PROCESSES  |
 ---------------
    |       ↑
    v       |
 ________________
 | TECHNOLOGIES |
 ----------------
```

Figure 3.1: Three Levels of Business Culture

The process structure, in turn, drives the technical structure. The technical structure contains all of the accepted routines, laws, and truths of the culture, regardless of their form. (Note: there was some debate as to whether the process structure drives the technical structure or vice versa. The issue was temporarily resolved by stipulating that the technical structure, regardless of where it came from, was basically an enabler of the process structure.)

The panel accepted as its problem definition the task of examining each of the three cultural levels to determine, from the perspective of the chief executive officer of an enterprise, how IRM was evolving. It did not bother to define IRM except to say that it was an enterprise-wide process for the management of information.

The panel began by evaluating the evolving enterprise value system. Most of the changes here were determined to be well known. After elaborating those value changes that seemed to have the most influence on IRM, the panel then debated whether to take on processes or technologies first. It concluded that process change was the most significant area of interest, but that technological change should be examined first. After doing so, it addressed the issues of changes that are occurring to the processes governing enterprise-wide IRM as a result of the noted changes in values and technologies. By far, the bulk of the meeting time was taken up in examining process changes.

3.1.1 The IRM Value System.

Using a brainstorming technique, the panel resolved that there were basically five major business trends that were affecting IRM in the enterprise. These were:

1. An evolving asset management mentality.

2. An increasing tendency to accept information technology as a significant influence on business strategy.

3. An increasing tendency of businesses to modularize themselves into small, distributed operating entities, generally referred to as " strategic business units."

4. A significant increase in computer literacy throughout the enterprise.

5. An increasing tendency on the part of corporations to substitute capital for human resources in an effort to increase the effectiveness of those resources.

3.2 THE IMPACT OF THE BUSINESS ENVIRONMENT ON INFORMATION TECHNOLOGY

3.2.1 Environmental Factors.

The business environment of the late 1980s will continue to increase in complexity and competitiveness. Successful firms in many markets will be those who can create and maintain global strategic capability and encourage and manage innovation. In addition, they will foster independence of functional and business activities while managing the necessary interdependences of these activities. There are common threads through these keys to success. More firms are beginning to realize that one thread is the effective use and management of information and its related technologies.

In many firms, computers have been viewed primarily as an operational support tool. In many cases, they have been used as a mechanism to control costs; however, in most firms they have been viewed as a cost to be controlled. Recent advances in information technology combined with innovative thinking on behalf of operational executives and managers, have led to uses of computer technology which have significantly increased the competitive capability of the company. Information technology and its associated systems are becoming increasingly vital components of a company's strategy to gain entry to a market, increase market share, or increase the switching costs for their customers.

3.2.2 Asset Management Mentality.

The business attitude toward information resources has changed. In many organizations, information resources (information, application, hardware, system software) have become embedded in the process of daily operations. Organizations become so dependent on some aspect of an information resource that interruption of access inhibits efficient business function. The critical role of these resources is forcing management to rethink its attitude toward planning for and managing them. They have become as important as human and financial resources.

Consequently, an asset management mentality towards information resources is emerging. Information resources have evolved from mere expense control mechanisms to assets "leveraging" the organization to more effectively meet its long and short term goals.

The move towards the asset management philosophy is forcing business organizations to critically review the creation, use, and disposition of information. They must identify how these assets are needed to meet organizational and departmental needs, manage how they are shared, and determine how to measure their effectiveness.

3.2.3 Information Technology Influences Business Strategy.

New information technologies have created new options for implementing and supporting a variety of business operations. For example, fresh approaches are needed to identify new products, manage and service current offerings, and review how both new and existing products are marketed and distributed. Due to this impact on operations, the management of information technologies is assuming a significant role in the strategic planning process. To support this effort, information resource measurements will become a more significant aspect of the accounting and control measurements for analyzing and understanding the performance of an organization.

3.2.4 Business Modularity (Strategic Business Units).

There has been a trend toward breaking organizations into modular units. This trend is related to the effective management of large organizations and their ability to respond quickly and strategically to changing markets. The issues for information resource management become:

- Understanding the need for shared information and technical resources.
- A clear perception of which information resources are needed for a particular unit.
- How shared information resources are to be created and used by various units.
- What information resource policies and standards are necessary to support the network of business units that form the organization.

3.2.5 Increasing Computer Literacy.

Business management is becoming increasingly sophisticated in its understanding and attitude toward information technologies. The increased understanding, whether personal or theoretical, is hammering away at the Jericho Walls of the high priests of MIS and DP. There is an increasing demand for ready access to information and a general understanding that changing technologies and related economics are making this possible.

3.2.6 Substitution of Capital For Human Resources.

The relative drop in the cost of information technologies has accelerated the shift of investment in human resources to capital investment. This shift brings down the total cost of running the business, and increases the effectiveness of those directly using or being supported by information resources. A growing number of businesses are beginning to experience this positive economic impact, much as the insurance industry experienced it in the 1950s, 60s, and early 70s.

3.3 TECHNOLOGY

After taking on the issue of enterprise values, the panel evaluated the area of information technology. Again, it used a brainstorming technique to identify all the technologies that it felt would be of significant interest. The panel produced a robust list of technologies and then attempted to evaluate their significance:

- Voice I/O
- Speech recognition
- Microcomputers
- Gigaflops
- Logical teleportation
- Communications standards
- Automated systems generation
- Natural languages
- Smart cards
- Huge bandwidth
- Smart telephones
- User friendly interfaces
- Adaptive systems
- Abstract data type machines
- Normalized application stores
- Data encyclopedias
- Home computing
- Professional workstations
- Video and audio technology
- Digital/logical computers
- Cyborgs
- Integrated media
- Biological computers
- Parallel processors
- Cross-language interpreters
- Communications standards with "portable" processors
- Computerizing the application development process
- Cheap storage
- CD roms
- Inference engines
- Database machines
- Robotics
- Heterogeneous DBMSs
- Intense international technology competition
- Very large scale integrated systems
- Digital representation of products and processes
- Reduced instruction set computers
- Graphics
- Function level firmware
- Image processors
- Transformers
- Fiber concentrators

The panel's initial objective was to determine whether or not there was a "personal computer-like" technology waiting around the bend that would have as dramatic an effect on the whole concept of IRM as did the personal computer. The panel concluded that there was no such technology.

It then attempted to identify any technological voids that it felt might inhibit changes in IRM processes or values. A void was determined to be any area where technological breakthroughs were required before a desired IRM process change could be accommodated. Again, it came up empty.

From the technology perspective, the panel concluded that, while technologies may not be assembled or tuned to perform all of the new process tasks that are anticipated for IRM in the 1990s, all of the technologies required to support the future environment exist, today, in some form or another. The basic task that lies ahead is to assemble and tune those technologies to the new management processes, once those have been determined.

3.4 PROCESS CHANGE

In starting to address the issue of process change, the panel stumbled on an interesting problem. This was a problem of how to define IRM as a process separate from other processes such as marketing, finance, product development, etc. currently operating in the normal enterprise. In one sense, it is important to distinguish the IRM process, and in another it is important not to distinguish it. It depends on the value system.

Recognizing this dilemma, the panel attempted to examine the IRM process as if it were a distinct management process that had distinct inputs, outputs, and controls, and as if there were some notion as to how to measure its efficiency and its effectiveness.

The panel decided that the efficiency of the process was measured from within the process itself, while the effectiveness of the process had to be measured from outside the process. This meant that efficiency could be measured, for example, by a programming supervisor who was monitoring lines-of-code-per-hour produced by his staff or an operations supervisor who was monitoring computer-resource-units, while on the other hand, effectiveness could only be measured by a user.

This concept meant that the user, per se, is outside of the IRM process. This idea creates problems for those who would like to treat the user as an integral part of the process. But, if we did that, we would have had no objective way of measuring improvements in IRM effectiveness. The best the panel could do was to allow the user to play two roles—one of the roles is inside the process and the other role is outside the process—and hope that the user himself could distinguish when he is playing which role. (Note: the panel agreed that the IRM-role is becoming much more dominant in the life of many users, and that this trend will continue until, for many of today's white collar users, it will be the only role they play. The problem of measuring their effectiveness, however, will become more difficult as their role changes.)

In examining the IRM process, the panel quickly determined two important ideas. First, the main outputs of this process are "application systems." Each of these systems, classically, contains its own inputs, outputs, and storage facilities, and it is uniquely designed to satisfy a fixed set of requirements. Each application system has its own life-cycle; that is, it is born, grows old, and dies.

The second important conclusion reached by the panel was that the classical IRM process which creates these application systems could be likened to the management process in a manufacturing job shop. This process is intended to create special, unique products, from scratch, one at a time.

The panel determined that the demand for information in the typical enterprise was becoming so complex and growing so rapidly, that the job shop management style that characterizes the current IRM process would have to give way to a new approach. This new approach to IRM would have to be based on what they called an "asset management mentality." In fact, the panel adopted the phrase "information asset management" (IAM) as a way to describe the main direction that they saw information resource management (IRM) evolving. (The panel even made its own joke: "I AM therefore IR.")

What did the panel mean by the word asset? Basically, it decided that an asset was any resource that was not consumed through use, i.e., any resource that was specifically developed for the purpose of being leveraged or reused in the creation of products or services.

The panel next decided that it had to provide a structure for asset management, and it proceeded to do so by defining what it believed to be the five basic categories of assets:

1. Assets employed in the acquisition of data.
2. Assets employed for the storage of data.
3. Assets employed for data manipulation.
4. Assets employed to produce information (reports) from data.
5. Assets employed to distribute data in any of the above modes.

Each of these categories was determined to describe assets because each of them was seen to transcend all applications. All applications must acquire, store, manipulate, report, and distribute data. If the applications were envisioned to be the vertical structure of IRM, then each of these asset categories was seen by the panel to be part of IRM's horizontal structure. See Figure 3.2.

The panel proposed that as IRM evolves into the 1990s, there will be a general shift in management emphasis from the current vertical perspective towards the horizontal perspective. The epicenter of this shift will be around the concept of data, as opposed to information. This notion is based on the logic that data, itself, is an information asset, i.e., a given set of data can be reused to create many different specific instances of information. This notion of data as an information asset can be dramatized by the idea that from 400 data elements it would be possible to create 400! instances of information. That's a lot.

\ Applica-tions \ Information Assets \	Accounts Payable	Accounts Receivable	Cost	General Ledger	Etc.
Data Acquisition					
Data Storage					
Data Manipulation					
Data Retrieval					
Data Distribution					

Figure 3.2: Information Management Structures

After agreeing on the five basic asset categories, the panel decided that it then needed to examine each of these categories from the management perspective. Based on the advice of its academic contingent, the panel defined the IRM management perspective to include our primary functions: planning, organization, administration, and control. It constructed another matrix (see Figure 3.3) for this phase of its deliberations.

\ Information Assets \ Management Processes \	Data Acquisition	Data Storage	Data Manipulation	Data Retrieval	Data Distribution
Planning					
Organization				As Is | To Be	
Administration					
Control					

Figure 3.3: The IRM Process-to-Asset Matrix

At this point, the panel decided to break up into small groups. Each group took one of the asset categories and examined it in terms of the four basic management functions, that is, each group studied a column of the matrix. The objective was to explain expected changes in IRM management *concepts due to the expected shift to an asset management mentality, i.e., what will the differences* be between the as is IRM process, and the to be IRM process.

The following are the actual reports submitted by each of the groups.

3.5 DATA ACQUISITION

3.5.1 Today's Problems.

- o Individual users plan acquisition independent of the enterprise.
- o No enterprise-wide prioritization.
- o Little formal scanning for external sources of data.
- o No formalized understanding of needs and association with sources.
- o Redundant sourcing—(inconsistent naming, identification, definition, etc.)
- o Different organizations entering the same data.
- o Authorized sources of data are not identified.

3.5.2 Acquisition/Planning.

AS IS	TO BE
o No formalized acquisition planning	o Annual information needs and source plan implies:
- Individual users plan acquisition independent of the enterprise	- Prioritization/budgeting (what will be acquired and what will not be acquired)
- No prioritization (enterprise-wide)	- Scanning for external data sources
- Little formal scanning for external data sources	- Enterprise data model exists
- No formalized understanding of needs and association of needs and sources	- Justification with regard to costs and benefits—considers media alternatives (paper, pictures, digital, etc.)
- Information systems plans do not adequately address alternative sources—alternative media	- Measurement of plan performance ("x" data/$ expended)
	- Support of total enterprise as opposed to individual applications
	o What data will/will not be acquired
	o Who will control data being acquired, e.g., Enterprise CIO, Dept., Individual

3.5.3 Acquisition/Organization—Staffing.

AS IS	TO BE
o Different organization entering the same data	o Enterprise Data Administrative function to assign responsibility to organizations for acquisition, maintenance, and integrity
o No organizational point of control, accountability	o DA to report to CIO
	o Development of Information Systems/User data acqui-

sition specialists in:

media
sources
technologies

- Data acquisition will be done by the user in the normal course of "doing business

3.5.4 Acquisition/Administration.

AS IS	TO BE
o No clear lines of responsibility or authority with regard to data consistency (logical) o No clear lines of responsibility or authority with regard to enterprise wide data acquisition technologies (physical)	o CIO responsibilities are required at every organization where there is a Chief Operating Officer (e.g., SBUs, etc.) (may or may not include DP operations, "Application Development") o CIO has ultimate responsibility/authority for establishing data consistency and control in: naming definition formats timing accuracy/integrity security o CIO establishes enterprise-wide data acquisition technology standards (for "physical integration") o Designation of authorized sources o CIO has responsibility/authority for inventory management, control, and evaluation of existing data

3.5.5 Acquisition/Control.

AS IS	TO BE
o Control is not centrally integrated	o CIO has to define the control mechanisms (standards and compliance processes) that will be required for Enterprise data—establish precedents for departmental data o CIO establishes internal audit organization to enforce controls

3.6 DATA STORAGE

The trends regarding the use and management of computing technology (discussed in Section 3.2) have a significant influence on how organizations manage the process of storing and maintaining their data resources. This influence on the storage and maintenance of data can be felt along three major dimensions:

1. The planning for data storage.

2. The necessary organization to support it.

3. Administration and control measures.

3.6.1 Planning.

The asset management perspective requires that in the planning for data storage, attention be focused on maximizing the return on investment in data resources. Operationally, that means the focus will shift from individual application data resource requirements to one where the whole enterprise represents the dominant perspective. Correspondingly, that implies a shift from short-term immediate project requirements to a long-term multi-project perspective. Therefore, greater emphasis will be placed on data sharability among the several enterprise-wide applications.

The greater influence of computer technology on business strategy affects the planning of data storage by requiring a closer relationship between individual databases and the requirements for supporting strategic business applications. A corollary is the need for integrating data storage planning to support corporate plans. The corresponding increased corporate dependency on the data resources will force the planning process to account for appropriate data integrity control mechanisms.

The modularization of enterprises into smaller business units will require that the data storage plan reflect the new reporting structures and data usage patterns. Most important, the data storage plans must permit data access according to the modular business structure, independent of data storage considerations.

The widespread use of computing technology and the corresponding increase in computer literacy motivates some changes to the planning for data storage. The plans must deal with larger amounts of data due to larger number of users, a more wide variety of user types, and new computer applications which in many cases reflect the trend towards substituting computing technology for human resources.

Besides accounting for increasing levels of user activity and data storage requirements, planning must provide for better definition of available data resources to enhance user awareness and access. Just as important, planning must also account for the linkage with external data sources.

3.6.2 Organizational Implications.

Viewing data as a corporate asset is likely to speed up on-going trends in how companies organize to manage data storage. The following trends are likely to continue:

1. Creation of DA (vs. DBA vs. DB analyst) to plan and administer data resources storage (DA, DBA will be more important) (reports to CIO).

2. Integrating DRM into end-user computing activities (i.e., backup and recovery of micro based data, privacy).

3. Line manager's function should include data storage considerations.

The breakdown of business organizations into smaller business units and the widespread use of computing technology throughout organizations will motivate the distribution of data administration, database administration, and database design across the organization. Correspondingly, the personnel roles of data administrator, database administrator, etc. will be played by different individuals in the different business units. In many cases, as discussed below, individual users and user managers will informally play these roles.

3.6.3 Administrative and Control Implications.

With an asset management perspective, greater emphasis will be placed on data sharing mechanisms to increase return on investment in the data resources. Due to the greater importance and sensitivity for some of the data, its categorization in terms of quality level, strategic importance, return on investment, and who is to physically and administratively control it (corporate, department, individual users), becomes necessary for effective management.

The importance of computing technology to corporate business strategy translates into an increased need for data availability and integrity control to ensure reliability. Also, data managers, in an effort to derive economies of scale, are likely to centralize planning and implementation of access to external data resources.

On the other hand, the modularization of business organization into smaller business units is likely to lead to distributed data storage administration to manage the data resources pertinent only to individual business units. The great increase in computer literacy, derived from widespread personal computing activities, has created a growing need for data downloading/uploading capability from/to corporate mainframes. This distribution of the data storage function has exacerbated the need for company-wide data management policies regarding data security and data integrity, and has created the need for further education of top managers, department managers, and individual users on data integrity and security implications.

3.7 DATA MANIPULATION

3.7.1 Applicable Assumptions.

As systems decompose into their elements of data creation/acquisition, data transformation/manipulation, data storage, data distribution, and data output/report production, the manipulation of data will be accomplished through the increasing development and use of functional processes and procedures which in and of themselves are assets to be used and reused across the enterprise, wherever applicable.

These functional processes will take the form of easily accessible and addressable "macros" which will accomplish standard solutions in areas such as math and statistical analyses, financial analyses, engineering design, manufacturing, etc. They will be rule-based functions that will be assembled to accomplish specific data transformation through linkage of existing processes and the addition of new processes if and as required.

The standard processes will have multiple implementations, e.g., on central processors, on intermediate processors, and micro or work station processors. They will be available throughout the network of processors, and they will be relatively processor independent, that is, a process will be implemented on multiple processors if the enterprise has multiple processors and on multiple vendor equipment.

The processes will be dictionary-controlled, that is, there will be a dictionary of available processes with their functional descriptions and relationships.

There will be neutral interfaces from process to process and from process to the data storage, *data acquisition, data distribution, and data production elements of these decomposed systems.*

The processes will tend to be organizationally independent, that is, wherever a standardized type of financial analysis is performed in the company, it will be performed using standard procedures and processes whether it is performed by the financial organization or not.

3.7.2 Planning.

The implications on IRM planning as a result of this approach to data manipulation and transformation will be substantial.

- o Planning must become more process- and rules-driven rather than activity-driven.
- o Planning for development must include planning for the reusability of any processes that are defined, rather than planning the development of processes that are recreated each time the process is required.
- o Process definitions must have functional orientation rather than organizational orientation and be standardized across all organizations. This implies the institutionalization of a coordination and agreement strategy for the development and publication of standard dictionaries of functional processes.
- o Planning must include and incorporate the maintenance of the fundamental business processes so that they can be adapted to accommodate changing technologies and alternatives, that is, as the enterprise changes equipment, changes interconnectivity, etc., and as the industry provides different alternatives for processing, the catalog of existing processes must be reviewed to determine appropriate implementations of each process. For example, an analysis of the design of a structure can be accomplished on a variety of machines with tradeoff of time and cost. As new equipment (i.e., vector processors or parallel processors) becomes available, existing analysis processes would be looked at to determine which ones would take advantage of the evolving technology.
- o Functional management must be involved in the definition of the standard processes, as opposed to, or in addition to, individual users defining those processes for themselves.
- o Planning must accommodate and acknowledge the condition of existing portfolios of systems which over time must be migrated to a process orientation that can be accomplished through

continuing modification and enhancements of existing portfolios.

3.7.3 Organizational Implications.

New roles must evolve within the enterprise, and it is not significant whether they are assigned to centralized or decentralized organizational responsibilities, although some tend to be enterprise-wide and some tend to be more user-related.

- To borrow from the artificial intelligence lexicon, "knowledge engineers" must be developed to abstract, from functional management and existing practices, standard process rules that are, in fact, enterprise-wide and appropriate for standardization and implementation as reusable processes.
- A role of implementation and maintenance of the efficient standard processes for use throughout the organization must be developed.
- A role of application assembling must be developed to link these reusable processes together, both to prototype new applications and perhaps to create the new functioning processes for data manipulation and report production.
- An additional role of application refining or tuning needs to be developed to transition the assembled prototypes to production, thereby insuring quality and efficiency of implementation, if and as appropriate.
- Since this catalog of reusable functions should be accessible to the general user base, the development of user support and consulting roles in the use of the standard processes needs to be developed.
- There needs to be considerable work in the area of defining environments and architectures so that decisions can be made as to what processes run best on what equipment throughout the enterprise.
- The role of maintenance must be enhanced to retrofit this approach to existing portfolios and to "mine" the existing portfolios for de-facto standard processes.

3.7.4 Administrative Implications.

This approach requires considerable education, training, and retraining since it shifts the whole development from a job-shop mentality to a continuous-process mentality, and the training must involve a good deal of business process education.

Considerable management activity and administration must be devoted to change management.

3.7.5 Control Implications.

Current policies, procedures, and standards need to be reviewed and redefined around processes, as well as around applications and organizations. This applies to areas such as how to measure performance of the organization against business plans and how the predefined processes can best be used to accomplish those plans.

- The procedures dealing with the security and integrity of the process itself, as opposed to merely the data. That is, does this standard analysis process indeed accomplish an acceptable analytical result?
- Intense control procedures need to be developed to audit the standard processes so that a user cannot modify the process to his own end without intervention by some other agency to insure the integrity of process and its resultant impact on enterprise data.
- Procedures and standards must be developed for rules development and maintenance.
- Configuration control of which version of a process appears in which sets of manipulations must be maintained.

3.7.6 Predictions.

Clearly, this approach requires and is based upon an assumption that computer literacy will be relatively high and widespread throughout the organization.

This approach enables and strongly supports business flexibility and modularity, as well as the ability to rapidly customize products and services. This is because the basic underlying functions of business are standardized and can be assembled appropriately for new products and services, rather than having to create entirely new application portfolios to accommodate changes.

This approach should speed the assimilation of technology into the enterprise, since it allows technology to be applied to portions of the existing portfolio. That is, individual functions can take advantage of new technology without the requirement for entire applications to be rewritten.

This approach should accelerate the substitution of capital equipment and processes for human resources since it frees the human resources to assemble existing processes rather than to recreate them for the nth time. By so doing, it frees intellectual resources from the repetitive task of recreating the predefined processes that are standard throughout the enterprise. It allows people more time to concentrate on those things that are unique and value additive for the enterprise, thus increasing revenue, reducing cost, etc., and to work at the point of the arrow of change rather than at the broad base of infrastructure.

3.8 DATA RETRIEVAL AND USAGE

The advent of the microcomputer and the growing power of the mini, combined with the concept of the data warehouse, led our group to consider the varying value of data at different organizational levels. Clear consensus was reached that data value is parochial. Some subset of data was of value to the professional, but not to the department as a whole. Some subset had value to the department but not to the corporation. This characteristic is depicted in the Venn diagram in Figure 3.4.

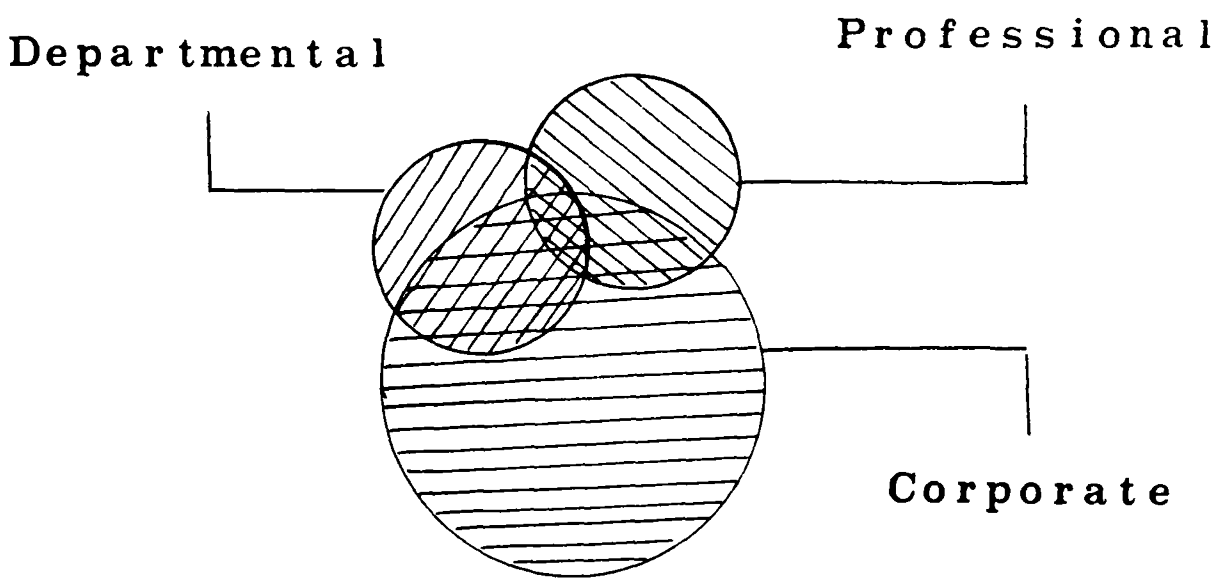

Figure 3.4: Data Usage

Although only two dimensions are represented, the model is more useful if three are imagined. The corporate circle then becomes a sphere, the department circle becomes a small set of cylinders, and the professional circle becomes a large number of thin circular slices intersecting both their department "slabs" and the corporate sphere. Data shared across departments and all professionals resides in the corporate sphere; departmental data shared among professionals resides in the cylinders.

The three circles suggest that data management for developing and maintaining the data resource must be practiced to varying degrees at three levels (for some organizations, this concept should be extended to reflect the number of meaningful levels). Our group believes that some form of value analysis will become increasingly useful in setting organizational structures, policies, standards, and procedures.

The concept of the intrinsic value of data has other facets. Not all data elements are created equal. The corporate sales number is more important than Harry's travel expense. Nor are the useful lives of data values the same. Yesterday's stock quotation is of limited importance to today's investor.

Achieving true asset management in connection with the information (or data) resources requires some metric, some means of valuation. This is necessary to assess costs versus benefits, to perform the make-or-buy analysis, and undertake appropriate insurance practices. While the group did not attempt to determine the metric (or metrics) necessary, it is clear that the cost of the related media upon which data is recorded is not a good one. Principal approaches may be to evaluate the cost of acquisition, storage, and purification, or to assess demand.

3.8.1 Trends.

The group analyzed how the "pushing" by information systems (IS) organizations and the "pulling" by other corporate organizations will affect the shape of information asset usage in the 1990s. Table 3.1 summarizes our collective vision.

	IS Organizations Will Push	Other Corporate Organizations Will Pull
Planning	1. IS organizations will undertake IAM training. 2. The development of common interfaces to users, to foster a single system image. 3. Information utilization as a means for better business operations. 4. Planning the corporate information resource.	1. Increasing micro literacy will accelerate the demand for corporate information. 2. Issues of: Availability Timeliness Accuracy Optimal Cost Security Reliability will surface.
Organization		End users will control much systems development directly through end user software.
Administration		Separate metrics for cost, price, and value will become necessary.
Control		Data Security Usage Metrics Value Metrics Cost/Benefit Analyses are essential.

Table 3.1: Summary of Trends

3.8.2 Conclusions.

The intensifying competitive environment American businesses face will engender the need for adaptable organizations. The information systems function will preclude the necessary corporate agility unless it positions itself more flexibly. Burying the corporate data asset in disparate software and hardware systems is a terrible inhibitor to flexibility, and is untenable in such an environment.

Fortunately, perhaps, the systems development process is being quietly usurped by end-user developers. Standard inquiry, report formatting, spreadsheeting, and graphics software provides the means to obviate arcane, mainframe mega applications. We expect the result to be the advent of a data utility—a warehouse where the corporate information asset is maintained. IS management emphasis will shift from the process of systems development to the planning, development, and ongoing care of the data resources and the assets related to their acquisition, storage, and dissemination. Focusing on these assets will result in standard technology platforms (tool sets), the better to leverage human resources as well as hardware and software within and without the IS function. An information asset management approach, complete with the appropriate metrics, will be essential to fully develop information resources.

3.9 DATA DISTRIBUTION

3.9.1 Definition.

Distribution is the movement of the information asset through the steps of data acquisition, storage, manipulation, and use.

3.9.2 Assumptions.

In discussing the distribution of data, certain assumptions are made. Only electronic information is included. All types of electronic information are covered. The generally identified types of information are data, voice, image, and video.

Although only electronic information is included, non-electronic distribution such as the mailing of floppy disks or magnetic tapes is considered part of the distribution environment. The distribution can also be external as well as internal to an enterprise.

Information Services will become embedded in the business with distribution of information an integral part of the business function. This will require a network orientation rather than a node orientation.

There will be function and data independence. Data will be viewed from the enterprise level rather than from individual functional areas. There will be a single conceptual model of all of the data of the enterprise, facilitating ease of access and management control.

3.9.3 Planning.

Plans will be developed for both the distribution of data and the management of that distribution. Management planning will address organization, administration, and control.

To manage the changing technological and contextual environment of data distribution, an architectural approach to planning will be required. The architecture will include both internal and external networks, where they are applicable.

Inputs

Inputs to distribution planning will be decisions from the planning of data acquisition, storage, manipulation, and use. All of these plans will be based on business needs as well as the existing infrastructure and anticipated technology changes. Research into future technology will become an important part of planning.

Outputs

The principle outputs will be the opportunities and constraints of the distribution technology, plus the overall distribution architectural direction.

Specific statements on the cost effectiveness of a distribution network must also be developed. The cost effectiveness must be structured so that it can be of assistance in business justification.

Implementation Plans

Implementation plans must address whether the network will be developed internally within the enterprise or if an external network will be used. This decision will influence the level and types of detail needed in the implementation plan.

The implementation plans must allow for flexibility in specifying technology, facilities, and sites.

Operational Plans

Operational plans will encompass the specific actions to be taken over a short time frame, normally one year or less. The operational plans will include budgeted amounts for implementation of the distribution environment. These amounts will be planned expenditures for the budget cycle. Actual expenditures will be compared with the budgeted amounts.

3.9.4 Organizational Implications.

The organization required to support the distribution environment should be in concert with the enterprise organization. If the business is highly centralized, then the distribution organization should be centralized also. If the business is decentralized, then the distribution organization should be decentralized.

In either situation, there should be a central management group for the distribution network, sometimes called the backbone network. The management of the nodes and the local area networks can be organized as a single entity or as autonomous groups, depending on the structure of the enterprise.

The organization should be so structured as to facilitate the interfaces between the plans of the enterprise, technical issues and opportunities, and the other areas of the data environment.

There is the possibility that the distribution network will become a utility within the enterprise. If this happens, the operational network will likely report at a low level within the enterprise. This reporting will not necessarily be within an information services organization.

Depending on the enterprise structure, facilities such as local area networks, hardware, and software may be acquired locally within established enterprise wide standards.

Staffing

There will be a need for a highly technical staff for the backbone network. Over time, these skill requirements may be reduced depending on the success of expert systems and other network aids.

The management of the distribution environment will need a strong business understanding. A direct relation to the success of the enterprise will need to be established.

3.9.5 Administrative Implications.

Administrative functions will include such things as billing, capacity planning, service quality, utilization reporting, and problem management.

In most environments of the nineties some form of billing of usage or service will be required. This may be similar to the way other utilities perform billing today.

Capacity planning will be at the network level where today it is mostly at the node level. Capacity planning will include decisions on the relocation of data movement, storage, and processing as well as on reconfiguring the physical facilities.

Quality will be judged by the satisfaction of business needs, as well as the accuracy and integrity of data within the distribution environment and the response time of delivery.

3.9.6 Control Implications.

Control will be an important component of providing quality service. Control will relate to the security and the integrity of data in the distribution environment. There will be established standards of performance for availability, quality, and data interchange.

3.9.7 Predictions.

The type of distribution facility will vary depending on the size of the enterprise. The principal distinction will be whether the network is an internal part of the enterprise or is an external environment.

Large enterprises, those with more than $20 billion revenues, will have large internal networks connected to external networks. Medium-size enterprises, from $1 to $20 billion revenues, will use more external networks with internal local area networks. Small enterprises, less than $1 billion revenues, will primarily use external networks.

3.10 CONCLUSIONS

At the end of the workshop, the panel attempted to draw some conclusions from its deliberations. There were several obvious conclusions, such as the idea that future information managers will have to develop a clear appreciation of asset management, and that before they can do so, enterprise management will have to modify its thinking to deal with information as an enterprise-wide asset as opposed a departmental expense. No new news. However, there were some not-so-obvious conclusions that emerged; and, even though there was not unanimous agreement among the participants on these conclusions, they are worth mentioning.

1. Applications will be data driven. It was generally agreed that the traditional idea of the application system, i.e., a system with its own unique approaches to input, storage, manipulation, retrieval, and distribution, will have to be replaced as the primary product of the enterprise IRM process. Rather than having computer applications as the primary deliverable, there will evolve a concept such as data applications. In the former case, the common denominator among systems is the machine they run on; in the later case, the common denominator is the data they share. This change in the idea of what the IRM process produces, i.e., what users get for their money, so to speak, will have dramatic and far reaching effects on both the current IRM processes and their supporting technologies.

2. The traditional systems development life cycle is obsolete. For some time there has been a general understanding that even though application systems seem to have a distinct life cycle, the rules that govern that life cycle do not seem to apply in the same way to data. Data does not die when its host application system dies. In fact, a great deal of time and money is spent today in the system development process attempting to salvage data from dying applications. Today's methods of systems development have no internal structure for developing and reusing any asset structure, not even data. Left to their own devices, projects will define requirements in their own way; they will ignore resources used for other projects; and they will produce stand-alone applications from scratch. There is no concept of integration among independently developed applications, nor is there any concept of asset building and reuse. An asset-based life cycle model will have to replace the current model.

3. Users will be a free market. The current approach to IRM is driven by the idea that users can define their own requirements. This in itself is a reasonable expectation. However, the idea of a requirement has come to mean different things to users and IS folk. The user has come to think that a requirement defines a need for information; whereas, the IS folk still think that requirements describe a total application. As applications continue to increase in size and complexity, application requirements become less meaningful to users, and they place more of a burden on them to attempt to define what they do in terms approaching algorithmic precision. The ultimate outcome of this tendency is to force the users to define the operations of the whole enterprise as an algorithm, just so that they can get some new reports. The economics of this approach are absurd. Therefore, users will be released from their obligation for defining application system requirements, and they will be allowed to create demands for information at will. The IRM problem will be to service these demands economically and efficiently—not to evaluate their reasonableness within the context of an enterprise-wide megasystem.

4. IS as we know it may go away. The current organizational form of the information services department, responsible for managing computer hardware, programming, applications development, applications maintenance, databases, systems software, communications, information centers, etc., may not be appropriate to an asset management environment. It may, in fact, evolve more toward a structure that organizationally accommodates the five—or whatever—asset structures. We have already seen organizational evolution toward those structures in the form of data administration departments (storage), communications departments (distribution), and information centers (output). With the increasing complexity of demand for information in most businesses, it would not be strange to see the vertical (applications) structure of IRM evolve in the same direction as general business management has evolved in response to the same types of pressures from its marketplaces, i.e., toward a concept similar to the strategic business unit. In other words, IRM could take on more of the shape of a business (or a set of business units) within a business—responding to pressures from what now appears to be a sort of internal information micro-economy maturing within each enterprise. Each of these business units would be focused on asset management.

The final set of tasks taken on by the panel involved the identification of what it believed to be the main inhibitors to the evolution of the asset management concept of IRM as we move toward the 1990s. It identified three primary inhibitors:

1. The current structure of information services. It was the general consensus that the current structure of IS has evolved as a bureaucratic roadblock to change. Existing vested interests and political structures, not to mention job definitions and pay scales, coupled with the inertia of the programmer corps, make it extremely difficult to make any significant changes in the environment. It will take top management visionaries with the constitution of Clint Eastwood and the political prowess of Henry Kissinger to make changes head on. Most will be content to go with the evolutionary cycle, trying not to rock the boat.

2. The current software legacy. Most of what we do in IRM today is constrained by the some 70+ billion lines of COBOL code that we have already implemented in the form of various, non-integrated application systems and inconsistent databases. Most technical and organizational innovations get lost in the noise of this installed base of software. It cannot be left in the dust as we attack the brave new world of asset management. It must be reckoned with.

3. <u>The current concepts of IRM financing</u>. Today, IRM is financed primarily through some sort of budget allocation logic among IS and the using organizations. This concept institutionalizes the idea that IRM is an expense. To treat IRM as an asset based management concept, new financing strategies will have to be developed for each of the basic IRM asset categories. This will be difficult because, today, nearly all of the financing strategies—including cost benefit analyses—are based around the idea of an application system. These strategies are <u>not</u> conducive to asset formation, and therefore tend to reinforce the idea the IRM should be run like a job shop.

4. IRM AND THE SYSTEM LIFE-CYCLE

Beverly K. Kahn and Salvatore T. March

CHAIRMEN

Biographical Sketches

Beverly K. Kahn is an Associate Professor of Computer Information Systems at the School of Management of Suffolk University. Previously, she was on the faculty at Boston University's School of Management. Dr. Kahn received her B.A. in Mathematics and Computer and Communication Sciences and M.S. and Ph.D. in Industrial and Operations Engineering from the University of Michigan. Her research focuses on the practice and effectiveness of data administration, and her methodologies for requirements analysis and database design are used by several major corporations and government agencies.

Salvatore T. March is an Associate Professor of MIS in the Management Sciences Department at the University of Minnesota. He received his B.S., M.S., and Ph.D. degrees in Operations Research from Cornell University. Dr. March is active in both research and consulting. He has published widely in the database literature, and is currently an Associate Editor for ACM Computing Surveys.

PARTICIPANTS

Roger Buchanan	James R. Lowery
Gary Berg-Cross	Volodymyr Lyczmanenko
John Carlis	J. David Naumann
James E. Cline	T. William Olle
James D. Funk	Gerard A. M. Otten
David Jefferson	Prashant Palvia
A. Milton Jenkins	Bill Selfridge
Mohammad A. Ketabchi	Steven Spewak
John Link	Jon C. Stonecash
Mary Loomis	John R. Weitzel

4.1 INTRODUCTION

The implementation of Information Resource Management affects the system life-cycle. Concurrently, an organization's system life-cycle methodology affects IRM. This interrelationship was the focus for this panel, which was closely related to the work of the previous Data Base Directions workshop [GOLDFINE 1982] (a bibliography is at the end of this chapter).

Prior to the Workshop, the panel members were asked to consider the definition and scope of IRM:

- Is IRM the establishment and enforcement of policies and procedures for managing the company's data (and information) as a corporate resource?

- Does it involve the collection, storage, and retrieval of data as a "globally" administered resource?

Due to the large number of issues addressed by this panel, and the large number of panelists, the panel decided to divide into several working groups. There was no preconceived notion on the number and topics of each group, so in a brainstorming session that included all the participants, 16 issues applicable to the panel were identified. Some of the issues were assumed to be covered by other panels, and were therefore eliminated from consideration. The rest were grouped into four major topics:

1. IRM and the Organization
   Group Leader: B. Kahn; Members: J. Funk, V. Lyczmanenko, G. Otten, B. Selfridge, S. Spewak

2. The Management of Change
   Group Leader: J. Carlis; Members: J. Lowery, A. M. Jenkins, J. D. Naumann, J. Weitzel

3. Metadata to Support IRM
   Group Leader: S. March; Members: G. Berg-Cross, R.Buchanan, D. Jefferson, M. Loomis, J. Stonecash

4. Methodologies, Tools and Techniques
   Members: J. Cline, M. Ketabchi, J. Link, B. Olle, P. Palvia

Each of the groups addressed its IRM topic from the points of view of costs/benefits, impact, barriers, and a definition of success.

4.2 IRM AND THE ORGANIZATION

This working group, recognizing that the "IRM in the 1990s" panel was exploring the future role of IRM, decided to address the interaction of IRM and today's business organization. (It turned out that the overlap of discussion was small.)

The thrust of the group's discussion concerned how to help an organization successfully implement information resource management. A formal definition for IRM was not developed, although working definitions resulted from the discussions. The issues addressed were:

- What is the best time to introduce IRM to an organization? What determines the receptiveness of the organization?

- What organizational characteristics affect IRM? How can IRM be tailored for a specific organization?

- What is the relationship between IRM, system development, organizational objectives, and business unit goals? How does IRM facilitate the achievement of organizational goals?

- What is the impact of IRM on system planning? What planning is needed for IRM or as a consequence of IRM?

- What is needed to ensure that systems support organizational objectives?

4.2.1 The Introduction of Information Resource Management.

Many organizations want to implement information resource management, but there is no consensus on what implementing IRM means—other than the very general objective of managing information as an organizational resource. The group made no attempt to develop a formal and detailed definition. The form and structure of the information resource management function and its objectives are organization specific, and there is no magical quick success formula. The information resource management endeavor is dependent on both internal and external factors.

A pilot project is often used as the first step in introducing IRM into an organization. (The selection of a pilot project is described in greater detail in Section 4.2.6). An important issue to address is when to introduce IRM—an organization does not want to miss the "best opportunity."

4.2.2 Organizational Readiness for IRM.

There are events in an organization that lead its management to believe that it cannot continue to follow traditional methods in its management of information. The following scenarios describe events that occur in many organizations—situations that indicate the organization's receptiveness to IRM. These scenarios are representative of the panel members' experiences.

The most common event is a costly, unsuccessful system project or a collection of unsuccessful projects that constitute a multi-million dollar loss. There are usually many excuses for failed system projects or cost overruns. IRM methods and procedures may prevent similar situations from happening again, particularly when the problem was due to inadequate planning, unclear or incorrect objectives/requirements, inappropriate design methods, or insufficient end-user involvement.

Most organizations have an excessive backlog of system projects, each competing for limited resources. The focus of IRM on business goals and objectives should lead to a more effective prioritization scheme. System planning under IRM could even lead to a different set of projects for development, projects that are more in tune with the organization's needs.

In a decentralized company, there may be a number of separate MIS or DP functions competing for resources and authority. If IRM can be perceived as providing a competitive edge by enabling one MIS organization to be more effective than another, then management should be more amenable to the changes that IRM would bring about.

The systems that are implemented are not the "right" systems. That is, they are not the ones most critical to the organization. This occurs either because there is no organization-wide system plan, or because the existing plan is no more than an implementation schedule independent of organizational goals and objectives. Project selection is often based on who speaks the loudest or what is considered "most interesting," rather than on organizational-wide requirements. Organizations need a rational way to select and schedule system projects. Perhaps IRM could be the vehicle.

A system requirement for integrated data or the sharing of data "owned" by another system is seemingly impossible to satisfy. There are many possible reasons:

- The location and form of this data may be unknown.
- Another system/business unit will not make the data available.
- The form of the data is not suitable for the new use, and an integrated view cannot be agreed upon.

IRM provides the global top-down perspective best suited for creating a shared data environment.

Organizational events may bring about the implementation of IRM. The organization may be undergoing a major business change, such as:

- Re-alignment of the business due to economic conditions, changes in key personnel, greater emphasis on corporate planning and strategy.
- New overall organizational structure due to divestiture, diversification, acquisition, or merger with another company.
- Introduction of a new product.
- New competition.

These events usually require that the organization have new information readily available. IRM can be the vehicle to provide this high quality information in a timely and cost effective manner. Additionally, systems in an IRM environment should be more flexible and adaptable to these kinds of business changes.

Many organizations are having difficulty coping with changing and improving technology, and with the proliferation of this technology. The acquisition and application of new technological products such as database management systems, communication networks, personal computers, and major software packages requires the careful planning and discipline afforded by IRM. The establishment of an IRM function responsible for technology transfer could avoid potential business disruptions.

In summary, the following events indicate that the organization should be receptive to the introduction of IRM:

- Costly, unsuccessful system project(s).

- o Excessive backlog of system projects.
- o Competing DP/MIS functions for scarce resources.
- o Systems implementation plan is not compatible with organization-wide requirements.
- o Minimal data sharing.
- o Organization is undergoing major business changes.
- o Difficulties in responding and using new technology.

4.2.3 Factors Affecting Information Resource Management.

The structure of the IRM organization—its scope, depth, and effectiveness—is largely shaped by three factors:

1. The business and economic environment.
2. The organizational culture.
3. The information systems environment.

These three factors may affect IRM simultaneously, possibly pulling it in different directions.

The Business and Economic Environment

The economic environment is of concern since the organization is required to think in terms of both *managing a resource—called "information"—and an economic doctrine centered around cost*-management. The organization is attempting to keep its overall costs down while still trying to achieve an amorphous goal of undetermined value—IRM. The organization, it is hoped, will be driven to concentrate on managing "information" as a corporate/organizational asset, an asset that can be a useful competitive element.

It is important to realize that IRM will be initially seen as a cost factor. The overall economic environment and the organization's economic health determine the monetary resources available to commit to endeavors, of which IRM is only one. The economic climate must not be hostile towards the commitment to IRM. In a booming economy or thriving industry, companies are more inclined to make the investment in IRM [GOLDSTEIN 1981]. During recession, most organizations become conservative and are averse to the "risks" of IRM; that is, to spending money on an activity with no short-term monetary payoff. When the economic climate deteriorates, the DP and IRM budgets are usually cut at least as severely as other resource management areas. One Fortune 500 Company needed to cut its IS/DP budget around a half million dollars—about the size of the data administration function's budget. The easy way out was taken and data administration was dismantled. With the elimination of data administration, all progress and effort towards IRM was lost. Unfortunately, this is not an isolated case.

The forecasts for the economy and the industry have similar effects on establishing or expanding the role of IRM in a company. Corporate financial condition has an impact on IRM. In a healthy company, IRM can establish credibility through vital long-range projects. In harder times, the lack of resources and the short-term orientation of organizational objectives can make it difficult for IRM to survive. In summary, the profitability and expected revenues of the organization play an important role in the willingness and ability of the organization to invest in IRM.

The Organizational Culture

In a recent article, Jane Linder [LINDER 1985] categorizes a number of characteristics of companies into five general dimensions. These dimensions provide a useful spectrum for describing the "culture" of a business. Figures 4.1 and 4.2 from this paper summarize her dimensions culture.

Another way to view corporate culture has been developed by Stevenson [STEVENSON 1985]. Stevenson identifies characteristics of organizations that are willing to seize opportunities and to commit required resources (to endeavors such as IRM). These organizations have an entrepreneurial focus, as contrasted to organizations with an administrative focus. The entrepreneurial culture is compared to the administrative culture in Table 4.1.

There are cultures or climates favorable to IRM, and others that are not. Favorable conditions are those that enable IRM to be established and to thrive. In an organization with characteristics unfavorable to IRM, the practice of IRM will be a constant struggle, and may ultimately fail despite its best efforts. There are profitable companies whose cultures are not favorable to IRM.

ENTREPRENEURIAL	CONSERVATIVE
Risk encouraging	Risk-averse
Informal	Formal
Decisive	Deliberate
Results-oriented	Process-oriented
Aggressive	Defensive
CLEAR LINES OF AUTHORITY	AMBIGUOUS AUTHORITY
Functional/divisional	Matrix
Profit and loss responsibility	Cost and revenue centers
Hierarchical	Consensual
COOPERATIVE	COMPETITIVE
Team-oriented	"Macho"
Collaborative	Individualistic
Reward-oriented	Censure-oriented
Merit-based	Power-based
LED	MANAGED
Long-term goals	Short-term goals
Clear, enduring mission	Mixed messages
Big picture-oriented	Detail-oriented
Created	Analytical
ETHICAL	AMORAL
Visible ethical policies	Tacit acceptance of unethical behavior
Ethical leadership and supervision	Hiring for cultural fit
Internal checks and balances	No attention to reconciliation between systems

Figure 4.1: Dimensions of Business Culture

FAVORABLE	UNFAVORABLE
Entrepreneurial	Conservative
Long-Range Goals	Short-Term Objectives
Cooperative Collaborative Mgt.	Dictatorial Power-Based Mgt.
Invests in New Technology	Fear of Technological Changes
Innovative	Slow Changing
Decisive	Bureaucratic
Strategic Positioning	Focused on Quarterly Profit

Figure 4.2: Organizational Characteristics and IRM

The placement of the IRM unit within the organization affects both its scope and its ability to focus. Its placement depends on the organization's Nolan stage and organizational culture. From DP/MIS failures, organizations have learned that it is impossible to develop a full system development from the requirements of the CEO (Chief Executive Officer) completely downward to the application requirements. A good organizational unit to start IRM in would be a business unit with sufficient local authority to set its own business objectives and goals. These goals should be specific—the more detailed the better. The goals and objectives should drive and prioritize all actions carried out as a consequence of IRM. The size and location in the organization of this business unit determines the suitability of creating superordinate as well as subordinate IRM units.

Another factor which influences the style of IRM and its potential actions is the planning and budgeting process of the business unit (which should be and usually is consistent with that of the whole organization). IRM often requires an organization to change its planning and budgeting horizon from short-term to long-term and lengthen the payback period for its IRM investment. The quarter to quarter profit mentality, characteristic of American companies, and bonus schemes related to those results, may be a hindrance to implementing IRM.

There are many ways to finance the IRM endeavor, with its placement in the organization both affecting and being affected by its financing. IRM may be considered an overhead expense, a special project of top management, or an expense allocated to the business unit. The manner of financing the IRM operation and its corresponding actions is based on the allocation of the financial responsibility of the corresponding business unit and the planning horizon, either long or short term, of top management towards the objectives related to IRM. These issues greatly determine the style of implementation of IRM. Usually, the process of implementing IRM is a series of IRM-related activities with intermediate successes.

	Entrepreneurial focus: CHARACTERISTICS	Entrepreneurial focus: PRESSURES	Administrative focus: CHARACTERISTICS	Administrative focus: PRESSURES
STRATEGIC ORIENTATION	Driven by perception of opportunity	Diminishing opportunities Rapidly changing technology, consumer economics, social values, and political rules	Driven by controlled resources	Social contracts Performance measurement criteria Planning systems and cycles
COMMITMENT TO SEIZE OPPORTUNITIES	Revolutionary, with short duration	Action orientation Narrow decision windows Acceptance of reasonable risks Few decision constituencies	Evolutionary, with long duration	Acknowledgment of multiple constituencies Negotiation about strategic course Risk reduction Coordination with existing resource base
COMMITMENT OF RESOURCES	Many stages, with minimal exposure at each stage	Lack of predictable resource needs Lack of control over the environment Social demands for appropriate use of resources	A single stage, with complete commitment out of decision	Need to reduce risk Incentive compensation Turnover in managers Capital budgeting systems

Table 4.1: The Entrepreneurial Culture vs. the Administrative Culture

	<------------------------------ Entrepreneurial focus		------------------------------> Administrative focus	
	CHARACTERISTICS	PRESSURES	CHARACTERISTICS	PRESSURES
		Foreign Competition Demands for more efficient resource use		Formal planning systems
CONTROL OF RESOURCES	Episodic use or rent of required	Increased resource specialization Long resource life compared with need Risk of obsolescence Risk inherent in the identified opportunity Inflexibility of permanent commitment to resources	Ownership or employment of required resources	Power, status, and financial rewards Coordination of activity Efficiency measures Inertia and cost of change Industry structures
MANAGEMENT STRUCTURE	Flat, with multiple informal networks	Coordination of key non-controlled resources Challenge to hierarchy Employees' desire for independence	Hierarchy	Need for clearly defined authority and responsibility Organizational culture Reward systems Management theory

Table 4.1 (cont.): The Entrepreneurial Culture vs. the Administrative Culture

The Information Systems Environment

The sophistication and complexity of the DP/MIS environment can also affect IRM. The information systems environment contains forces that indicate the level of cooperation between users and management on one side, and the IRM/DP organization on the other. One widely known scheme for describing this environment is the Nolan Stage Theory [NOLAN 1982, 1979], outlined in Table 4.2.

Clearly IRM encompasses stages 5 and 6 of the Nolan Model. Indeed, IRM may be the mature "sixth stage." IRM must be evolutionary and not revolutionary to have the best opportunity for success.

An evaluation of the current business state of affairs will provide input for the determination of the Nolan stage for each business unit. Those units at the highest stage will be most appropriate and receptive to IRM. The introduction of a very tightly controlled/controlling IRM unit will never be accepted in the early stages, stages 1 to 4. The style of the IRM unit and the way it interacts in the different business operations is closely related to the Nolan Stage of the business unit. During the later stages, a tighter controlled and more sophisticated management information will be required to evaluate the business unit with respect to its business goals. An evaluation of the stage of the various activities in the business organization will be required, and determines whether a more loosely or tightly controlling IRM is most suitable.

STAGE	PURPOSE	APPLICATIONS	DP PLANNING AND CONTROL	OBJECTIVE OF CONTROL SYSTEMS
1. INITIATION	Computer Acquisition	Functional for cost reduction	Lax	None
2. CONTAGION	Intense System Development	Prolifer- ation	More lax	Facilitate growth
3. CONTROL	Prolifer- ation of controls	Upgrade doc- umentation and existing applications	Formalized	Contain supply
4. INTEGRATION	Shared data	Upgrade to database technology	Tailored	Match supply and demand
5. DATA ADMINIS- TRATION	Promote data adminis- tration	Integration of applic- ations	Shared data and applic- ations	Contain demand
6. MATURITY	Steady state	Integration mirrors information flows	Data resource strategic planning	Balance supply and demand

Table 4.2: The Nolan Model

4.2.4 IRM and Organizational Objectives.

Information resource management is primarily instituted to aid the whole organization achieve its objectives, and secondarily to benefit DP/MIS. Therefore, the thrust of IRM is: how can it aid the organization to better formulate its objectives and goals in such a way that progress, in achieving the set goals, can be better controlled. Objectives are linked with specific goals, and goals can be measured. IRM helps to clarify the organizational objectives and to formulate consistent business goals with respect to these. Additionally, IRM makes the systems development unit a more integral part of the organization. This leads to the development of the "right" systems so that organizational objectives can be more easily achieved. This assumes that the organization has long range objectives. Some organizations do not have long range objectives, and IRM can still be a valid objective.

How Can IRM Help Clarify Corporate Objectives?

To be successful in today's competitive environment, corporations must have clear, unambiguous business objectives at all levels of operation. The alignment of these business objectives requires that businesses integrate and share data vital to the continuing effective and efficient utilization of company resources. To be effective in supporting the business, information technology must be applied in such a way that it properly supports these business objectives.

An organization utilizing information resource management produces an integrated systems plan. This consists of a prioritized list of systems and databases which become candidates for implementation. The plan is specifically designed such that important business objectives are fully supported and that maximum advantage is taken of data sharability. The IRM process acts as an integrating factor between the business and information technology. A major premise of IRM is that computers should be used to support the business. Therefore, the analysis and identification of the business and its objectives should be done before there is any concern for computer solutions.

An important foundation for IRM is the Business (or "global activity") Model. The purpose of the business model is to provide a view of the activities of a business, to identify important information flows, and to act as the basis for analyzing business objectives. Another important IRM deliverable is the Data/Information Architecture, which identifies the information necessary to conduct the business. The business model and the data (information) architecture are linked: business activities process the information defined in the data architecture.

Business requirements are therefore described in terms of business processes and the information they use. It is premature to consider computer solutions to business problems without having a firm grasp of the business and its needs. If computers are to be used effectively, let alone as a strategic weapon, then the business and its objectives need to be clearly defined at the outset.

The business model consists of a set of charts, at varying levels of detail, showing business functions and the information flows between them. Using the principles of top-down decomposition, each function at a given level is "broken down" into more detail at lower levels.

A business model is particularly useful in that the knowledge of the information needs across the business is developed. The business model is a valuable communications tool because often no single individual has a complete knowledge of how the business works. Most importantly, the business model is a functional, and not an organizational, representation of the business. Organizational structures are subject to change, and functions are frequently replicated across an organization.

As part of the IRM process, the mission and objectives of the business are defined by those responsible for the strategic direction of the business. Using the business model, sub-objectives can be defined for each function/business unit such that their contribution to the overall objectives can be clearly seen. This process helps in the formulation of a realistic set of objectives with an appropriate degree of measurability and specificity.

Defining and analyzing business processes provides a comprehensive understanding of how the business meets its objectives and accomplishes its mission. Analysis of the decision-making processes creates a basis for distinguishing among strategic, tactical, and operational processes.

Finally, IRM helps clarify business objectives by emphasizing the provision of quality data (from both internal and external sources) at a reasonable cost. The timeliness, accuracy, consistency, and cost of data is dependent on a properly designed information environment such as that advanced by IRM. Additionally, IRM fosters intra- and inter-business unit communication.

In summary, IRM helps to clarify organizational business objectives by providing high quality corporate data that can be used to:

- o Portray a corporate-wide view of information.
- o Define and reformulate objectives and goals so that they are actually measurable.
- o Verify that business unit goals are consistent with organizational objectives.
- o Measure business unit goals and organizational objectives.
- o Establish the responsibility for achieving business goals and objectives and the monitoring of progress.

How does IRM Affect the System Development Process?

There are plans created as a result of the IRM process which provide benefits to the organization. These plans, which reflect a consistent view of business objectives and information requirements, are based on a better understanding of the business and its objectives and, because the plans are derived principally by business management, provide a greater consistency between the expectation and delivery of computer solutions.

Therefore, if information technology is to be successful in supporting organizational objectives, those business objectives must permeate down to and affect the design of both systems and databases. Too often however, objectives are set at the very top of an organization, while the computer systems design is done near the bottom, ignorant of these objectives. The IRM process affects this issue by ensuring that systems planning is directly driven by the business objectives.

IRM explicitly shows the linkage between business objectives and systems development. IRM therefore encourages communication; systems development knows exactly what contribution is being made to the business. The business end-user knows what is being done by systems development to support his business, and hence develops more realistic expectations. System development priorities are derived from business objectives that were defined and agreed upon by the business managers themselves. *System development therefore has an objective measure of system priorities.* System planning is part of the organization's strategic and tactical planning process.

Because the IRM process deliberately encompasses a wider scope of the business than a typical stand alone project, IRM supports the creation of an integrated systems environment. Applications *do not stand alone, but share data with other applications. The IRM process addresses issues of* data sharing, meaning and consistency across various application areas and organizational units. Working from the "architectural plans" produced by IRM is easier than having to make individual, isolated decisions at the systems development project level.

The IRM process generates and is supported by an "information encyclopedia" (or "metabase") of architectural information which becomes an important reference source for system development. The content of this meta-data is described in Section 4.4. The first step of any system development project is always a feasibility study, followed by requirements analysis, which involves determining the way the business operates, its need and information requirements. The IRM architectures already contain much of this information, therefore facilitating and shortening requirements analysis.

Under IRM, organizational resources are better utilized. Replication and redundancy in systems development are minimized and systems efforts are directed at activities that are high priority from the organization-wide perspective. The data architecture is of major significance in guiding systems development. It precisely defines many business rules and resolves previously unclear or inconsistent concepts which often lead to lost productivity in systems development.

Finally, the IRM architectures are not technology-bound, and hence have the potential for defining information flows across all types of systems—office information systems, end-user computing, and standard information systems. IRM encourages the acquisition of new techniques that improve system development. These include data-driven development, data modeling, prototyping, and fourth generation languages.

4.2.5 IRM and Planning.

Information resource management affects how the organization accomplishes system planning. Aditionally, IRM planning usually encompasses system planning. IRM planning consists of two phases: 1) planning for the establishment of an IRM function and 2) planning for the development of an information architecture. Today, the broader term "architecture" is used instead of the term "model" used extensively in [GOLDFINE 1982].

The first phase concentrates on strategic issues and might be called a Strategic Information Systems Planning (SISP) study. Before the SISP study is undertaken, a plan is developed. This plan is the basis for selling IRM to top management. Top management is sold on the idea that for a limited commitment, the initial and potential benefits of IRM will be readily apparent from the results of the SISP study. After the SISP study, IRM is more than an amorphous concept. In a study [KAHN 1983], it was determined that management's lack of support for IRM can be attributed to its lack of understanding of what it is and what its effect is.

The SISP study is usually completed in six months or less. It is intended to be minimally obtrusive and disruptive to the organization. Three main deliverables are produced in a SISP study:

1. A description of the "gross" information architecture.
2. A cost/benefit analysis of implementing the information architecture.
3. A plan for developing an information architecture.

The SISP study concentrates on the organization's information architecture.

Many tasks are involved in the production of the information architecture. Most of these must be done in the order described:

1. Analysis of Business Objectives.
   Top management is interviewed to establish the organization's business objectives and goals. Full cooperation and mutual commitment of top-management and the SISP team is essential. Business objectives are precisely and consistently documented.

2. Global Data Modeling
   The global data model is an enterprise data model. It documents what data is required to support the business objectives. Data is described in terms of high-level aggregates representing the most vital concepts, ideas, and things for this organization and/or business unit.

3. Business Modeling (also called Global Activity/Function Modeling)
   This task is accomplished from the business unit perspective. Given the Business Objectives documented in (1) and the Global Data derived in (2), one determines the major activities needed to support these goals and produce/maintain the data. The global activities are high-level functions; often, there are one or two global activities for each business unit. Often this task is done from the business unit perspective. Each organizational objective is decomposed into goals for each business unit.

4. Data/Activity Cross-referencing
   This task determines the consistency between the global data model and the business/global activity model. Through matrix manipulation, one can derive data and activity clusters. Each cluster supports one or more vital business objectives.

These first four tasks are driven by the fact that the primary objective of IRM is to help support the organization's business goals and objectives. These tasks are similar to those in IBM's Business Systems Planning (BSP) [IBM 1981, DAVIS 1982].

5. Assessment of technological needs and directions
   It is necessary to determine and evaluate the overall technological hardware and software directions of the organization. This verifies that current business and system objectives are supported, and provides a plan for the future. It may include the study of technologies such as: external databases, database machines, office automation, expert systems, distributed processing, networks, etc., as long as the technologies are clearly required to achieve business objectives and help to make information technology a competitive asset.

6. Assessment of personnel skills and organizational structure
   It is necessary to determine the available and required personnel resources (skills) to make IRM happen. Additionally, the current organizational structure needs to be evaluated with respect to its ability to foster and support IRM. Necessary education and training, as well as position, scope, and tasks of the IRM function should become clear. This is sometimes called a human resources architecture.

7. Construction of the information architecture
   From tasks 1 through 6, the gross information architecture is constructed. Several alternatives are produced. Selection by top management will be based upon the corresponding cost/benefit analysis and development plans.

The last two deliverables will be described below, and form an integral part of the results of the SISP study.

The SISP study is a highly interactive process with extensive top management support and managerial involvement. The mode of operation is prototype-oriented; that is, there are many checkpoints for correcting the scope and direction of the study. The documents produced should not be a shock to any of the participants. It is essential that the SISP team (6 to 12 full-time people plus interviewees) complete the whole SISP study within 4 to 7 months. This timetable and resource commitment has proven feasible in business units with up to 15,000 employees.

The second deliverable of the SISP study is the cost/benefit analysis. This analysis indicates which business objectives can be better or more easily achieved, or produce more profits through increasing revenues or decreasing costs as a result of the information architecture and IRM. These potential benefits are detailed. The necessary investments for the implementation of the information architecture can be offset by these cost savings.

The third deliverable is the actual plan for the next period (usually 2 to 5 years) of IRM implementation. The plan concentrates on the gradual development of the information architecture. Obviously, this planning is driven by the priorities of the business objectives/goals agreed to by top management but moderated by technological and organizational influences (as analyzed and understood by all parties). Decisions that were traditionally and solely DP/MIS are now the joint responsibility of business management and IRM/DP-experts. Additionally, this plan includes precedence analysis of the organization's application portfolio and DP activities.

Through this approach the implementation plan is better synchronized with business objectives/goals and results in greater compatibility between expectations and products (e.g., systems). The system developers now better understand business expectations and at the same time the "business people" have a good understanding of compatible system architectures within and between business units. In this way business requirements are better formalized and potentially can be integrated, thus ensuring a better utilization of human and EDP resources. Additionally, all system projects are consistently and correctly prioritized and an overall cost/benefit analysis is completed.

4.2.6 Ensuring that Systems Support Organizational Objectives.

The success of IRM and DP/MIS totally depends upon the ability of its systems to help management achieve the organization's objectives and goals. A collection of actions must take place during the development of such systems. First, business objectives should be clearly and consistently formulated and communicated throughout the organization. This is the responsibility of management. Subsequently, the whole business unit must work hard to fulfill the established goals and objectives. Information should reflect the business in such a way that achievements can be measured to verify goal accomplishment.

Every organization is a complex organism attempting to reach continually changing targets. All portions of the organization need feedback and motivation to ensure that everyone is directed towards setting and achieving common goals. Success in IRM can be defined as supplying the necessary feedback to operational levels of management about their performance against the goals set by their superiors, and supplying the feedback to top management about achievements of the business units. Finally, the satisfaction of the information users about the service provided by IRM/DP to them is the last element of feedback to close the circle.

After the SISP study has been completed and management has decided to go with the further implementation of IRM, the actual implementation of the information architecture begins. The selection of the pilot project is the first task. Proper selection of a pilot project is crucial to the success of IRM. This project should have the following characteristics:

- The project's (sub)systems must be small enough to complete relatively quickly, but must be of sufficient size to be considered a real system and not a "toy".
- The business area covered must be substantial, but not so vital as to endanger the business unit if the project is delayed or unsuccessful.
- The area should be free from politics.
- The introduction of new technology or system life-cycle methods/techniques should be avoided.

A successful pilot project is very important, but not essential for IRM to proceed and succeed in the organization. Care should be taken to document the events that occur throughout the pilot project, so that subsequent projects can learn from the pilot's successes and failures. IRM may subsequently succeed with a poor or unsuccessful pilot.

In summary, IRM helps systems development and systems themselves achieve organizational goals by:

- Providing a model of the business that shows the information flows between business, and improving/encouraging inter-business unit communication.
- Creating an information architecture of the organization's business units across all technology.
- Providing a technology architecture which encourages the acquisition of new techniques that improve system development.
- Providing better utilization of organizational resources and documenting the use of these.
- Providing unification of plans—consistency between organizational objectives, business unit goals, and system development priorities.
- Providing better feedback, especially to top management, concerning the progress towards achievement and eventually, the successful achievement of organizational objectives, business unit goals, and systems projects,
- Providing upper management the means to evaluate and refine its goals and to ensure that they are measurable.

- Providing a discipline/charter for standards and procedures that should save money in the system life-cycle.

4.2.7 Conclusion.

IRM affects the whole organization. It is the glue that links organizational objectives with systems and technology. It facilitates the development and evaluation of organizational objectives and the formulation and execution of consistent business unit goals.

The organization with IRM views information as a organizational asset that deserves and requires management. The organization sees IRM and information as a strategic weapon contributing to its success.

IRM makes "the systems organizational unit" part of the organization, rather than an outside entity providing a service in the manner of an independent company. A unit that feels part of the organization is more likely to put out the extra effort that is often required.

4.3 THE MANAGEMENT OF CHANGE

Information Resource Management is a strategy for managing change. Management of the organization's information resources is expected to improve responsiveness to the organization's information needs, at known costs and with known benefits. The task group approached the "management of change" aspect of IRM from two directions: management of change <u>under</u> IRM, and management of change in <u>getting to</u> IRM.

While the benefits of IRM involve increased responsiveness to change, there are associated costs and barriers. Among the costs are:

- The need for centralized metadata creation, maintenance, and dissemination.
- The need for multiple systems development methods in decentralized development organizations.
- An organizational climate that welcomes change.

Barriers to implementation of IRM include:

- Individual and organizational inertia and resistance to change.
- The need to retrofit the existing applications portfolio.
- The lack of tools to support both metadata and applications development based on the metadata.

Successful IRM could be identified and measured by:

- Reliance on centralized metadata.
- Presence of a Chief Information Officer (CIO) in the organization.
- Increasing IRM budgets.
- Decreasing central application development budgets.
- Diminishing application development in the central IS organization.

4.3.1 Introduction.

This section first discusses the issues involved in migrating towards IRM. We define a starting point thought to be typical of today's large organizations, and suggest the activities such organizations must accomplish as they move toward IRM.

The next section discusses the management of change, both in general and as change is related to IRM. The objectives in dealing with change include minimizing the impact of change, and improving responsiveness to it. A number of strategies are suggested that, together with IRM, help meet those objectives.

Part of the impetus towards IRM is the need to support a user community that is increasingly aware of its information needs and increasingly capable of implementing its own technological solutions. User developed systems and the Information Center are clearly related to IRM. The next section discusses these issues.

The task group summarizes its report with a discussion of the costs and benefits of IRM, the barriers to IRM, its expected impact, and the means of measuring the progress of organizations towards IRM.

4.3.2 Migration Toward IRM.

The migration to IRM is contingent on the starting point. Further, a range of starting points undoubtedly exist in organizations today. So a few words describing the starting point assumed in this section is necessary. These assumptions are:

- o The CIO resides at a senior management (not executive) level.
- o The MIS organization maintains (has custodial responsibility for) a large applications systems inventory/library (representing a large capital investment) that is written in a procedural level language and is (on average) over 7 years old.
- o Most of the existing data resources reside on non-relational DBMS and sequential-type files.
- o PCs exist in the user environments.
- o At least one 4GL is utilized for MIS applications.
- o The systems development staff is largely centralized.
- o User-developed systems exist.

To migrate to IRM in most organizations, the following actions will be necessary:

1. Establishing an information architecture—a database of databases. IRM requires the ability to integrate data across databases efficiently.
2. Upgrading existing DBMSs. IRM requires flexibility, compatibility, and rapid access.
3. Establishing and populating an active data dictionary. IRM requires improved efficiency in the development and maintenance of application systems, for which control over data definition is essential.
4. Determining data ownership (application system ownership as well). IRM requires a clear and enforced taxonomy of data ownership. Corporate (organization) data must be distinguished from departmental or personal data.
5. Retrofitting existing application systems. IRM requires the efficient and effective use of the raw material (data), and its efficient transformation into information.
6. Reorganizing the MIS/IRM organization. The organization's structure must support the new orientation—service, rather than control—and the broadening scope of operations (telecommunications). The CIO must attain executive status (indicating corporate management's understanding of information as a resource) to acquire the financial support necessary for long-term payoff activities.
7. Exploiting existing hardware and software technologies and tools. IRM requires huge increases in the productivity of systems personnel. This means automation where possible and the providing of computer support where automation is not possible.
8. Educating client management in information resources management. IRM requires managers outside the MIS organization to be capable of managing information.

4.3.3 The Management of Change.

The notion of the management of change in IRM connotes a proactive attitude toward change. It involves an investment in anticipating change, planning for it, and implementing it. The forces of change come from the business environment and technology.

Because IRM is charged with the overall management of the information resource, it must manage the acquisition or creation of information, its storage, maintenance, use, and its disposal when it becomes obsolete. Changes in technology (i.e., in computer hardware, computer software, telecommunications hardware, telecommunications software) change how these functions in the life-cycle of information can be performed.

Because the system life-cycle is intended to translate a need for a solution to a business problem into a useful information system, it must accomplish problem analysis, solutions synthesis, and solution implementation. Changes in technology (i.e., in DBMSs, data dictionary systems, 4GLs, application generators) change how these functions in the system life-cycle can be performed.

Changes in the business environment require that new information be managed (i.e., entry into a new business, new government regulations, new methods of operations all may require new information). Information about the environment itself and the changes to it may also need to be managed. Changes in the business environment trigger the system development lif-cycle to build the systems which handle the new information.

Planning

An organization can plan for responses to change in a number of ways. It is not unusual to plan for anticipated changes. It is unusual to plan for unanticipated ones, but it can be done by a 'futures' group, by emphasizing the impermanence of job roles, by inculcating an acceptance of the pace and inevitability of change. An organization that expects technological change will be the one most likely to succeed in abandoning old technologies and adopting new ones. In this case, systems and IRM will be evaluated on how easily they can be modified.

Information as a Resource

Information should be managed differently if it is considered a resource rather than an expense. It is unlike other resources (money and people) in that it is not consumable (although it does perish); it can be copied (but the copies may become inconsistent). When information is considered a resource the focus is on business rather than technology; on what rather than how; on information cost and benefit rather than data gathering and storing. There is less technical emphasis, but only because it is supported by better technology.

Information as a resource should affect how systems are developed. Data will not be owned by an application but by the organization. The choice of systems to develop should be on a more business-like footing. Cost effective use of information includes the dismantling of obsolete systems, non-redundant data entry, and the control and reuse of data definitions (and perhaps program modules). When information becomes viewed as a resource, the organization's hierarchy changes to reflect the changes in scope, orientation, and power. The names will change from Data Processing to MIS to IRM or to other names which reflect the changing mission. The nature of the mix of job skills will change too, as a business outlook becomes more important than technological competence. It is ironic that technical competence will create a de-emphasis on technical competence.

Objectives for Managing Change

There are two basic objectives for managing change. First, IRM and its system life-cycle need to minimize the impacts of changes. Second, they need to find ways to increase their ability to deal with change so that they can be more responsive to it.

Strategies for Dealing with Change

We are suggesting several strategies for dealing with change to accomplish the stated objective.

- Make change a constant. Information resource managers must recognize that change is inevitable. They must assume that any given state of affairs with respect to technology or the environment will change—probably sooner than expected. Obviously, some things will not change, but since we do not know which they are, we must be prepared for changes in everything.

  Information resource managers need to institute systematic (although not necessarily formal) scanning of the environment outside of the information resource functions. The purpose here is to look for indicators of change both to technology and to the environment. Technology scanning clearly lies within the purview of IRM. Environmental scanning must be carried out in conjunction with the organization planning/scanning function (again systematically but not necessarily formally).

  IRM should plan to make systems (information processing or information distribution) and existing technology obsolete. For example, when a particular type of technology is adopted, an event or type of event should be defined which would trigger the search for a replacement technology. When a system is implemented, a date could be set for the reassessment or replacement of the system.

Perhaps we need to summarize these points by saying that IRM should keep moving—avoid lack of movement for protracted periods of time to avoid stagnation. Management of change must be proactive rather than reactive. The idea of keeping moving leads to the next strategy.

- Break long range plans into short term projects. When change is pervasive and accelerating, long range plans will become obsolete before they can be carried out. Therefore we suggest that small is beautiful. Small projects, executed quickly (perhaps in 1 year or less) in a series permit delivery of tangible results. As each project is completed, the overall plan can be reassessed and modified as necessary. It is much easier to change direction from project to project, than it is to change a five year plan when it means starting all over.

- Subdivide information architecture. Databases and systems should be designed as "independent" entities so that changes to one system or database will have minimal impact on other systems or databases. This is not to be interpreted as a return to uncoordinated development. It suggests that information processing systems and information distribution systems be clustered around common databases. It requires that standardized interfaces be defined so that systems in different database clusters can pass information back and forth. It also requires metadata about where to find types of information.

- *Develop flexible methods. Flexible methods require understanding of general/basic functions* in the system life-cycle. These would include problem recognition, problem identification, problem definition, solution definitions (also known as requirements specification), process and data analysis resulting in input and output requirements, system designs and functional *requirements, logical system designs, physical system designs,* decomposition to subsystems, decomposition to normal procedures and computer procedures, procedure design, program design, procedure writing, program coding, testing, user verification, and conversion. This may be the wrong list, but we believe that some basic things must take place to move from identifying a problem that needs solving to the creation of an effective system which solves that problem. Given that information resource managers can define these basic things, various techniques and tools must be used to produce a customized implementation of the basic development process. Which techniques and tools are used depends on the problem and the context. There is no one best way to produce solutions. Given appropriate tools and a previously defined database, prototyping may be appropriate. Given a well defined existing system, subsystem by subsystem replacement may be appropriate. The key here is to stay flexible and apply the right tools and techniques to the problem and to recognize that new tools and techniques are continually emerging.

  Finally, at least where there is some minimum level of understanding of the problem, the methods used should produce results quickly. If they are right, so much the better. If they are wrong, that will be discovered quickly and therefore corrected quickly.

- Reduce the size of changes. By staying in motion, taking change as a given, anticipating change, and keeping projects small within the context of a larger plan, the impact of change is reduced by reducing the size of changes. It is easier to assimilate sequential small *changes—each one causing a minor jolt to the existing overall system/organization—than it* is to cope with a massively disruptive major change. This is true of hardware, of operating systems, and versions of DBMSs. Subdividing the information structure also minimizes the impact of change by isolating it to a specific part of the system/organization.

Making change a constant and breaking long range projects into plausible small, short range projects increases IRM's ability to deal with change by causing information resource managers to deal with change on a regular basis. Thus, they stay in practice at dealing with various agents of change. Developing flexible methods increases responsiveness by speeding up the solution process. The impact of these approaches to change will be to develop a higher level of ability to cope with and manage change.

The costs include additional training, less efficiency in managing information, probably replacing hardware and software before it is fully depreciated, and the costs of learning and mistakes.

The benefits include more effectiveness in managing information and less risk of displacement by more innovative users of information.

The obstacles include the inability to see the need to change constantly, the inability to justify the costs, and the *unwillingness of IRM professionals and users to let go of old skills.* Measures of success include change transparent to users, and change timely enough to maintain or enhance an organization's position relative to its environment.

4.3.4 End User Computing and the Information Center.

Basic Premises

Persons involved today in data processing, management information systems, and information resource management must recognize the fact that prototyping, fourth generation languages, end user computing, personal computing, and information centers are handling the corporation's data in ways very different from the way it has been handled in the past. Furthermore, the ways those data are handled in the future may be very different from the ways they are being handled today. Therefore, the question which must be addressed by anyone working in the area of IRM is "What changes can be made today to one's IRM 'superstructure' such that it can be responsive to whatever ways the corporation's data and information are used in the future?"

By their nature, prototyping, end user computing, information center computing, etc., have been external to the traditional "system life-cycle." However, they should not be allowed to exist outside the purview of a broadly-defined view of system development. There is nothing which precludes that broader view from encompassing user-developed systems and even manual systems, many of which deal with corporate data and information.

Users doing computing at the departmental and/or information center level are affecting existing systems by discovering and applying new uses of the corporation's data and information resident in those existing systems. Likewise, end users (who are now interacting with the data in existing systems in new ways) are affecting new systems development as they discover new understandings of the corporate data and as they then seek to apply new uses to that data.

All of this leads to increased rates of systems development, which in turn leads to a faster pace of change within those parts of the organization that deal directly with system development processes. If rapid change in the area of end user computing is not managed effectively, either IRM will be left behind to be replaced by something new which can manage and respond to rapid change in this area, or IRM will find itself managing only a portion of the corporate data and information resource.

<u>The Impact of End-user Computing on IRM</u>

System development has begun to take place outside of traditional system life-cycle methodologies. Either no formal "life-cycle" is used or a very shortened, often abbreviated, one is used. If IRM is a part of the system life-cycle, persons involved in IRM must decipher what changes IRM must undergo in order to function within the constraints of these shortened or non-existent system life-cycles. IRM must be flexible enough to be able to deal with each end user development process differently, as dictated by the players and the organizational structure. User computing and information center interaction with user departments is by nature diverse and not holistic as is the case with most MIS-based functions within an organization. IRM must recognize and build upon that diversity.

Impediments to managing information arise at the end user computing and information center levels in the form of end user use of corporate data once those data have left the purview of an IRM-based "data model." Are there new data models for this now user-based subset of the corporate data? Should there be? At what point does corporate data cease to be "corporate data" and become "user data?" Ironically, it is this very dilemma that in many instances is the greatest push to begin managing information as a resource.

The corporate data model becomes much more visible to, and crucial for, the end user who begins to work more closely with "downloaded" or "subseted" corporate data, as he or she becomes more and more involved in end user computing. The end user begins to see the value in taking a more active part in validating the corporate data model and in managing the corporation's information in general. If this does not occur, one may be faced with an end user who is using decentralized data to make corporate decisions but who is not willing to have that data be a part of the corporate model.

<u>Some Ideas/Questions on End-user Computing</u>

If, as the working definition of IRM specifies, IRM "involves the collection, storage, and dissemination of data as a 'globally' administered and standardized resource," then what is IRM's future role as more of that corporate data collection, storage, and dissemination is occurring at the end user/departmental level without information center involvement and by end users who are not using traditional (or any) system life-cycle processes?

If systems development is taking place in end user departments, how can IRM be structured so that there can be a smooth flow back-and-forth and a balance maintained between what IRM needs from the end user's "system" and what value IRM can be to the user involved in end user computing? The rapidity of change makes it crucial for this interchange to be flexible and adaptable to different end user situations.

When prototyped systems are being built by end users, at what point during the dynamic prototyping process can/must IRM concepts and corporate IRM requirements be put in place for those prototyped systems? Must they fit the corporate data models as they are being built? At what point are changes made to the corporate data model based upon the dynamic prototyped system?

If it is assumed that system life-cycles do not apply to purchased or packaged software systems in the same way that they apply to internal system development, at what point should IRM become involved in purchase and/or installation decisions for such packaged software to ensure a high degree of fit between the product and the corporate data model of the organization and its end users?

4.3.5 Summary.

We have treated IRM as an approach to dealing with data and information that will improve the organization's ability to deal with internal and external change. Two objectives to meet in dealing with pressures for change are to minimize the impact of change, and to improve the organization's ability to respond.

Today's organizations do not have IRM. The costs of implementing IRM are very great, and include organizational change, centralized planning and control at least of metadata, and an eventual retrofit of the applications portfolio.

Forces tending toward IRM implementation include more widespread recognition of the value of managed information resources as the organization copes with its environment. Technology, especially the increased availability of processing power, is the primary impetus for IRM. As more individuals and units in the organization gain the ability to process data, the need to manage information just as other resources are managed keeps growing.

A number of factors inhibit the implementation of IRM. Probably foremost is the resistance to change that is part of organizations and individuals. To overcome this inertia, the move toward IRM must be impelled by the organizational climate established by the CEO.

In addition to resistance to change, there are a number of technical barriers to IRM. Tools for information architecture and for the management of metadata are not yet available (although under widespread development). The question of what to do about the existing applications portfolio, which must certainly be retrofitted to support IRM, has not been resolved. Systems development methodologies do not have the power and flexibility to support integrated applications.

There are some indicators of movement toward IRM. Organizations should be able to assess their progress by these measures. First, if the organization has established the position of Chief Information Officer, there is an indication that information has been recognized as a resource that needs management analogous to the management of other resources employed. Second, the presence and use of metadata tools such as central data dictionary/directory systems indicates that the organization is identifying and building control over the information resource. Third, a steadily increasing trend in expenditure for the information resource management function indicates recognition of the need for and value of IRM. Fourth, better management of the systems development process, indicated by fewer large projects, points toward better management of the data upon which applications are based.

4.4 METADATA TO SUPPORT IRM

4.4.1 Introduction.

This section develops a framework for understanding the information resource so that it can be managed. We have taken a top-down approach by focusing on "metadata." Metadata is simply data about data within a scope of interest. Our scope of interest is Information Resource Management.

The metadata for IRM is broadly defined to include both the data and process perspectives of an enterprise. It encompasses not only operational computer-based applications and data but also the system development processes. The scope of an organization's metadata is determined by its IRM program. It is important to note, however, that the concept of metadata for IRM is much broader than the level of descriptive data typically found in a data dictionary.

The next section describes why it is important to define and collect the metadata for IRM. Section 4.4.3 then discusses the representation of metadata and presents a set of orthogonal dimensions in which to describe IRM metadata. Section 4.4.4 discusses the effects of this framework on the system life-cycle. Section 4.4.5 offers directions for research and development.

4.4.2 Benefits of Metadata.

Before information can be managed as a resource, it must be understood. While this appears to be trivially true, it has been much neglected in practice and is, in fact, a difficult task. Historically, applications have been developed using bottom-up, process oriented approaches. The semantics of the data and functions of these applications have been buried in computer code. Hence they are unmanaged and unmanageable. The development of metadata for the information resource will help to standardize <u>what</u> data describes the information resource. <u>How</u> that data is collected is the concern of data planning and system development methodologies. It is not addressed here.

An important characteristic of IRM is that metadata is managed. Figure 4.3 classifies and interrelates the benefits of managing IRM metadata. This classification is not meant to be exhaustive, but represents a framework for thinking about what metadata must be represented to achieve benefits.

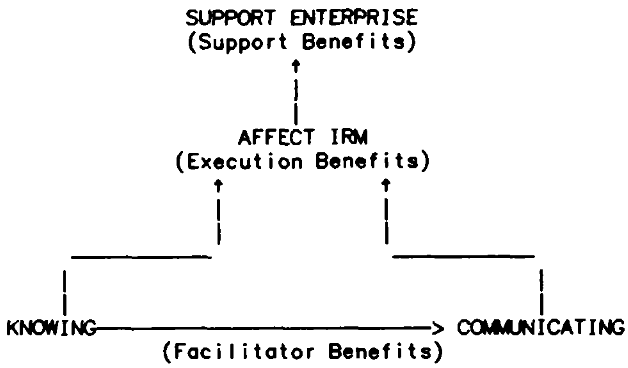

Figure 4.3: Benefits of Metadata

Facilitator Benefits.

The foundational level of benefits, which drives all the others, is the Facilitator level. Metadata facilitates knowing and communicating about information resources. Knowing must precede communicating. Metadata defines what it is that we are trying to manage. It provides a standard for data gathering, defines when development may proceed from one activity to another, and establishes a vocabulary for talking about data.

Execution Benefits.

Once we have defined what we need to know and have a vocabulary for talking about it, then we can manage information using IRM concepts. Direct benefits of metadata are obtained in the execution of IRM. Metadata permits us: (1) to define what it means to manage the information resource and (2) to develop a coherent set of tools to support IRM.

Without a well defined set of constructs to define IRM data, support tools are disjoint. The tools we have today for information and application planning, system development, data administration, etc., are typically very narrow in focus and are not well integrated. Metadata management can facilitate the integration of application-development data, as well as the integration of the information processed by applications.

Metadata supports IRM not only because it formalizes the information resource, but also because it formalizes policies by specifying what data must be collected as systems are developed and used. Metadata can be applied to support information systems planning, database design, data and process creation, maintenance, control and distribution. Issues such as security, integrity, reliability, and project management are also within the scope of IRM metadata.

Establishing a metadatabase provides a repository of data describing the development and operation of applications within an organization. This data should be captured throughout the system life-cycle, providing an effective tool for project management during initial development and for configuration management during later stages.

Support Benefits.

Given a repository of data about the evolving information resource, the needs of the enterprise can be better met. System development is supported because the metadatabase facilitates management of development data. Furthermore, data and processes can be more easily shared among applications because this development data is similarly maintained for all applications. Duplication of effort in data collection and data processing can be eliminated through shared data and processes. Many applications will be reduced to simple queries and database transactions (perhaps transformed by software into a series of physical database accesses). Therefore, in many cases data processing can be more responsive to user needs.

4.4.3 Representation of Metadata.

The metadata for IRM is complex. To facilitate understanding and communication, we propose a framework based on "dimensions" of metadata. These dimensions are motivated by five reasons:

1. Control of usage: defining who can do what with which data and when.

2. Abstraction: hiding details to reduce complexity.

3. Insulation: managing change by decoupling mechanisms.

4. Standardization: defining interfaces to facilitate interchange and sharing.

5. Communication: facilitating common understanding.

The dimensions of metadata are based on previous work in database standards, particularly from [OTTEN 1985].

We consider four dimensions: Type-Occurrence, Data Independence, Service, and Time. Each dimension is discussed below. We present the "points" within each dimension in sequence beginning with the business orientation and proceeding to the data processing orientation.

Type-Occurrence Dimension.

Different people are concerned with different levels of "meta-ness" of the information resource, hence this dimension, which classifies levels of data description. Four points along this dimension are appropriate.

- Application Data: the actual data and processes required to meet the users' information requirements. End users retrieve and update this data, and use these processes.

- Dictionary: defines the types of data and processes represented in Applications Data. End users refer to the Dictionary for data item and process names; the DA/DBA staff update the Dictionary.

- Data Model: defines the constructs (i.e., types) used to create the Dictionary. The DA/DBA staff refer to the Data Model to update the Dictionary.

- Fundamental Constructs: defines the basic constructs (i.e., types) used to create the data model. A Dictionary System vendor may provide different Data Models which can be interrelated via the Fundamental Constructs.

Note that these points are interdependent, since each higher level point in this dimension is the type description of the immediately preceding point and is the set of occurrences for the immediately following point. Type changes to one point require instance level changes at the immediately following point, except for the Fundamental Constructs which must be "self-contained" or self-defining.

Data Independence Dimension.

Each point in this dimension represents an independent perception of the "same" data or process. The scope or extent of the data/process perceived, its format, and the existence of derived data may vary independently. Having different perceptions of the same data minimizes the impact of changes in one perception or another. Five points in this dimension are appropriate.

- Presentation: describes the format of data as presented to the user independent of its storage or even logical representation. Time-series data, for example, may be logically represented as a table, but presented to the user as a graph. Similarly, an application process may be logically represented as a Pascal program, but presented to the user as a set of structure charts. The same data/process may appear in multiple presentations.

- User View: describes the scope of data (including derived data) and processes perceived. One user may, for example, perceive only department data and department budgeting processes, while the database actually contains department and employee data and processes exist to support a wide range of department and employee related information needs. Further, some users may perceive the result of some process as data. For example, "average employee salary" may be perceived as a descriptor of each department while, in fact, a process calculates average employee salary from the employee data. The same data/process may be included in many user views and thus be shared across users and applications. The data or processes in a user view can have many presentations. A user-interface facility should be responsible for performing the User View-Presentation mapping.

- Conceptual View: describes the full scope of data and processes within the organization. It is independent of any database management system or process modularization (software design). It represents the Information Resource Manager's view of the information resource. It may include both existing and planned descriptions. An enterprise has only one Conceptual View; all User Views supported must map to the Conceptual View.

- Logical View: describes database schemas and software designs. Logical perceptions are typically limited in scope (compared to the Conceptual View). They are dependent on the database management system and other system software used. They reflect design decisions about logical data storage and software modularization. A relational database schema and a system structure hierarchy are examples of Logical Views. The same data/processes may be included in multiple Logical Views (this typically implies redundancy). All Logical Views must map to the Conceptual View. An enterprise's data resource may be implemented in many databases, each described by a logical view.

- Physical View: describes physical storage and implemented processes. Data files and software are examples of physical perceptions. The same data or processes existing in different physical perceptions imply redundancy. Modifying a physical view, e.g., to improve system efficiency, should not impact the corresponding Logical View. A database management system should be responsible for performing the Logical to Physical mapping including both the data (e.g., physical record and file structures) and the processes (e.g., interpretation or compilation of a high level query language).

Service Dimension.

Each point represents the extent of orientation to business processes as opposed to data processing. This dimension reflects the need for various layers of services and interfaces. Four points are considered.

1. User Services: describes the services provided directly to the user and is reflected in user interfaces. Application programs, user queries, menus, screens, etc., are examples.

2. Tool Services: provides support for User Services by transforming requests for user services into system- recognized requests for data and processes. Data dictionary systems, query language processors, and report generators are examples.

3. Database Services: provides the logical access mechanism for system tools. It determines logical structures of data and software and how they should be accessed. Database Management Systems provide Database Services (e.g., by transforming queries into physical I/O requests).

4. Basic Services: perform the actual physical accessing and processing. Operating systems provide Basic Services (e.g., I/O routines).

Time Dimension.

The time dimension of metadata is continuous. It represents a chronology of events. It is the easiest of the dimensions to state, but is perhaps the least understood. Time is used to distinguish versions of data and processes. Since it is orthogonal to the other dimensions, each of the examples described has a time aspect. For example, there is a time aspect to software releases, backup and recovery of data, transaction management, system daemons, project management, etc.

4.4.4 Effects of Metadata on the System Life-Cycle.

In general, the effects of metadata dimensions are to focus attention on key issues and to suggest opportunities for improving the System Life-Cycle (SLC). The following addresses some of the dimensions and points that are particularly relevant for various SLC stages.

Strategic Systems Planning Stage.

An understanding of the need for metadata promotes the recognition of the need for an automated Dictionary to provide effective metadata management. This suggests the need for two tasks to support the Type-Occurrence Dimension:

- A miniature SLC to acquire a Data Model.

- A miniature SLC to acquire a Dictionary.

These are necessary in order to support a third task:

- Extension of the Dictionary by means of the Data Model.

The extended Dictionary will then be used to support the other stages in the SLC.

Business Analysis and Design Stages

The requirements of these stages include modeling, analysis, and documentation. Application of the Dictionary to support these requirements emphasizes the Data Independence Dimension. The following tasks are particularly important:

- o Documentation of User Views.
- o Creation of Conceptual View based on User Views.

Construction and Installation Stages

The time and cost of these stages place an emphasis on minimizing new development (buy rather than build) and the need for integration of components. This suggests the need to analyze the Service Dimension to:

- o Purchase Tool Services wherever appropriate.
- o Construct User Services based on tool interfaces.

Usage and Maintenance Stages

Efficiency and effectiveness during these stages requires an increase in control of the information resource and a reduction in the impact of change. Control of the information resource may require that security, privacy, and integrity constraints be expressed in the Dictionary (Type-Occurrence Dimension) and enforced by Database Services (Service Dimension). The impact of change may be reduced through the use of the Data Independence Dimension. For example:

- o Effects of conversion to a new DBMS may be isolated to the Logical point.
- o Effects of conversion to new hardware may be isolated to the Physical point.

Also, there may be a need for the Time Dimension to control versions of programs and data.

Evolution and Phaseout Stages

These stages have requirements similar to the Business Analysis and Design stages (e.g., modeling, analysis, and documentation), and also require the incorporation of new data and systems into the information resource. They need the Data Independence Dimension to support the following tasks:

- o Modeling of new User Views.
- o Analysis of impact on Conceptual View.
- o Analysis of impact on performance at the Physical View.

There is also a need for the Type-Occurrence Dimension to support major changes such as the following:

- o Extension of the Dictionary by means of the Data Model (to more easily model new aspects of the business or new types of data).
- o Extension of the Application Data by means of the Dictionary.

4.4.5 Conclusions and Directions for Further Research and Development.

There are many open questions in metadata management. The following is a brief list of some of the issues that arise from our framework:

1. How should each dimension be represented? How should each point in each dimension be represented? Do we have the right dimensions/points? What are the appropriate inter- and intra-dimensional interfaces and interdependencies? How can we map across dimensions? What's common? What's hard? What techniques should be used (data model, activity model, dynamics model, organization model, etc.)?

2. What are the requirements to manage each dimension/point? What are the impacts on tools/techniques?

3. What are the measures of effectiveness in each dimension/point? At least two measures are apparent: (1) the degree to which our knowledge about the information resource can be represented, and (2) how well we can manipulate that representation to solve business problems, e.g., to generate or maintain user applications.

4. What are the measures of efficiency in each dimension/point?

5. What opportunities are there to apply "expert systems," artificial intelligence, and knowledge representation technologies to metadata management? At least two possibilities exist: (1) diagnosis—scanning the metadata to discover problems and opportunities within the metadata (e.g., inappropriate uses of data and processes, opportunities for sharing data and processes) and within the enterprise (weak financial controls, opportunities for sharing facilities), and (2) metadata access—determining what metadata is needed by individuals within the enterprise and determining the best way to access that data.

6. How does the availability of metadata affect the traditional System Development Life-cycle? Improvements should be obtained in all areas due to a standardization of application development data, and the potential for sharing among applications and application development teams.

7. Are the dimensions/points organization independent? How do the size, industry, geographical location, etc., of an organization affect metadata management?

8. What skills are required to manage metadata?

9. Are the dimensions/points time-invariant?

10. Can we establish the relative complexity of data in each dimension?

11. What are the appropriate methodologies for development/use of metadata?

12. What is the impact of collapsing a dimension? Are all dimensions necessary? Are the above dimensions sufficient?

13. What is the appropriate set of Fundamental Constructs for the type-occurrence dimension?

14. What is the state of the art in metadata management?

15. What are the ramifications of having metadata encompass both the data and process perspectives of an enterprise?

16. What is the appropriate scope of metadata integration within an organization?

17. Are conventional database management tools suitable for managing metadata? If not, what additional capabilities are needed?

18. How should metadata be managed in a distributed environment (including distributed data, processes, hardware, system development, and data administration)?

4.5 METHODOLOGIES, TOOLS, AND TECHNIQUES

The objective of this group was to determine what methodologies, techniques, and tools are required in an organization's system life-cycle to support information resource management. This is a continuation of the work done in the previous Data Base Directions Workshop [GOLDFINE 1982]. Time was spent in defining terminology, defining the stages of the system life-cycle under IRM, identifying the types of techniques required to support each life-cycle stage and determining how to choose a methodology or methodologies and their underlying technique(s). No evaluation of specific products was undertaken.

The system life-cycle can be broken into stages which may vary depending on the methodologies used. However, for purposes of this discussion, the following list of basic SLC stages is proposed:

1. Strategic system planning.

2. Business analysis.

3. Design.

4. Construction.

5. Installation.

6. Usage and maintenance.

7. Evolution.

8. Phase-out.

The distinction between the business analysis stage and the design stage is felt to be of major significance, and the breakpoint between the two is a cause of frequent confusion. The work in a business analysis stage is, by definition, "descriptive" of a given area of the business, and in a design stage it is "prescriptive" of a system to support part or occasionally all of an area of the business already covered in a business analysis stage.

The breakpoint between design and construction is also one to be identified carefully, particularly when prototyping tools are being used. In a more traditional environment, the design and packaging of application programs would be regarded as part of a construction stage.

The following definitions were used to distinguish between methodology, technique, and tool: A methodology is an organized approached for handling part of the SLC. A methodology may consist of one or more techniques. A technique is a means of accomplishing a task in the SLC. For example, an entity-relationship diagram is a technique to accomplish the task of data modeling. A tool is a software package which supports one or more techniques.

4.5.1 Categorization of System Life-Cycle Methodologies.

There are three ways of categorizing a system life-cycle methodology (SLCM). In the first place, it is possible to categorize an SLCM according to the stages it covers. Second, it can be categorized according to the techniques used. Third, an SLCM can have a perspective.

Some cover the first three stages, some cover only design and construction, while others cover only a single stage. No one SLCM adequately covers all stages.

It is recognized that there exists a very large number of methodologies. Some cover only a single stage of the eight identified above, and few if any attempt to cover the whole system life-cycle. It is also noted that many of the available methodologies tend to emphasize the use of a somewhat limited set of techniques.

The SLCM may have a perspective such as data-oriented, process-oriented, or event-oriented. This perspective determines the suitability of techniques.

A good SLC methodology should have the ability to:

- Ensure integration with other systems.
- Ensure data standards.
- Address all eight stages of the SLC.
- Ensure controlled access to shared data.
- Ensure analysis of data and data relationships, with emphasis on constraints to be satisfied.

The SLCM should ensure integration with other systems. The automated business processes, the associated information, programs, files, etc., should be analyzed in such a fashion as to ensure that an understanding of, documentation of, and physical/logical development of systems is integrated into the current portfolio.

The SLCM should also ensure data standards, not just naming conventions, characteristics, but truly understandable and recognizable business names used across all systems.

All eight stages of the SLC should be addressed. At present, to achieve IRM objectives, it requires the use of several SLC Methodologies which, it is hoped, carry a high degree of continuity. To facilitate an even greater degree of continuity and to span all phases of the SLC, the SLC as it exists will have to be expanded and preferably automated to help design systems from the top down. This type of SLC would cover not only the current SLCMs, but be expanded to cover strategic system planning, business analysis, etc., from the top down and continue through to the phase-out, enhancement, etc., of each system.

The SLCM should ensure controlled access to shared data. As IRM inherently stresses integrated automated systems design and development, this also precludes the necessity for shared data. As systems in the past have been developed independently with little or no data sharing, the need for controlled access was not of importance as with IRM and shared data. As such, the IRM SLCM must ensure controlled access of this shared data (generally subject databases).

The SLCM should ensure analysis of data and data relationships, with emphasis on constraints to be satisfied. To ensure the maximized integration of systems and the usage of shared data, the SLCM must take care to ensure that a correct data model of the corporation is built and maintained. This is generally accomplished through entity analysis. Then as each system is designed the SLCM must ensure that the data constraints and user view synthesis (process model) are integrated into the data model, in line with corporate objectives.

4.5.2 Relevance of each SLC Stage to IRM.

The first two stages of the system life-cycle, namely strategic systems planning and business analysis, are felt to have an extremely high relevancy to information resource management. It is felt that if the stages are not carried out adequately, then the chances of achieving the aims of information resource management are minimized.

The importance of the other six of the eight stages of the system life-cycle was found to be either of medium or low relevancy.

The overall picture is summarized in Table 4.3. This breakdown indicates the importance of adopting the right kind of methodology at the very first stage of the system life-cycle.

STAGE	RELEVANCY
1. Strategic Systems Planning	High
2. Business Analysis	High
3. Design	Medium
4. Construction	Medium
5. Installation	Low
6. Usage and Maintenance	Low
7. Evolution	Medium
8. Phase-out	Low

Table 4.3: Relevancy of each SLC Stage to IRM

It is questionable whether the objectives of information resource management can be achieved by attempting to introduce the concept with systems that are at a later stage in their life-cycle.

4.5.3 Factors Affecting Selection of SLC.

Every organization must select the most appropriate system life-cycle methodologies. There are organizational and environmental factors that affect the organization's choices. These should be considered from an IRM perspective. The following factors were identified as being the most important in influencing the SLCM choice from an IRM standpoint. No attempt was made to prioritize these factors. This remains an issue for more discussion and research.

- o Designer skills available. The designer refers to the individual responsible for any part or phase of the system life-cycle.
- o Business objectives.
- o Availability of tools.
- o Organizational characteristics. This factor includes organizational philosophy, size, and industry type.
- o Maturity of organization with respect to data processing. A possible measure of this is the position of the organization on Nolan's six stage hypothesis.

- o Competitive influences.
- o Application area. Application area refers to the functional area and/or focus of the application on particular management layers.
- o Influence of user on management.
- o Degree of decentralization.
- o Hardware availability.

Each of the above factors can assume discrete values along a continuum. The values, to be called factor levels, need to be defined carefully.

4.5.4 IRM Contingency Matrix for Choosing SLCM.

The factors, described in the previous section, should be the guiding criteria for the final selection of SLCM(s). At present, no single SLCM addresses all stages of the system life-cycle; so more than one SLCM may be required. We envisage the development of a contingency matrix which will aid in the selection of appropriate SLCM(s). The matrix will list the independent variables (i.e., the organizational and environmental factors, and their levels) in the leftmost column and the dependent variables on the top-most row. The dependent variables are the SLCMs themselves, either existing, in development or to be developed. An outline of the contingency matrix is presented in Figure 4.4.

DEPENDENT VARIABLES		SLCM 1	SLCM 2	SLCM 3	...
INDEPENDENT VARIABLES					
	LEVEL 1				
DESIGNER SKILLS	LVL 2				
	LVL 3				
	LEVEL 1				
BUSINESS OBJECTIVES	LVL 2				
	LVL 3				
	LVL 4				
	. . .				

Figure 4.4: IRM Contingency Matrix

There are a number of system life-cycle methodologies in the marketplace today. Researchers may classify them into categories based on similar characteristics. Further research will result in filling in the body of the above matrix. Each cell of the matrix will contain data regarding the interaction of the SLCM with the independent factor level. At a minimum, the cell may rank the SLCM's applicability as Good, Fair, or Poor. In addition, qualitative and quantitative information regarding costs and benefits may also be presented in each cell. As an example, the cell may look like Figure 4.5.

PROS AND BENEFITS: ... CONS AND COSTS: ... RANK:

Figure 4.5: Example of Contingency Matrix Cell

One effort in building the contingency matrix is under way [PALVIA 1985].

Once the contingency matrix is developed, it can be used in a qualitative manner in choosing the appropriate SLCM(s). Another possibility is to assign weights (i.e., prioritization) to the independent variables and arrive at a quantitative measure for each SLCM under consideration.

4.5.5 Techniques Available for SLCM.

A major contribution is the identification of generic techniques that support all stages of the system life-cycle. Since the term "technique" was felt to need further clarification, a list of typical techniques was compiled (Table 4.4). In addition to listing the techniques, an indication is given of the SLC stages in which they might be used. It must be emphasized that there is no implication that these techniques have to be used. It is clear that many of these techniques are applicable to only one of the SLC stages, but some could usefully be performed in more than one.

GENERIC TECHNIQUE	RELEVANT SLC STAGE
Top-down data modeling	Strategic systems planning Business analysis Design
Bottom-up data modeling	Business analysis Design
Business activity analysis	Strategic systems planning Business analysis
Analysis of computerizable processes	Design
Data flow in the organization	Business analysis
Data flow in automated system	Construction
Business event triggering	Business analysis
Business event precedence/ succedence	Business analysis
Real-time system event triggering	Design
Material flow in the organization	Business analysis
Organization activity analysis	Business analysis
Screen format design	Design Construction
Program Prioritization	Design Construction
Access path analysis	Design Construction
Program cross-reference with data	Design
Transaction traffic analysis	Design
User work pattern analysis	Installation
Cost benefit analysis	Phase-out Business analysis Strategic systems planning
Performance analysis	Evolution Design

Table 4.4: Generic Techniques for SLC Stages

While opinions differ considerably about the relevance of some of these techniques to information resource management, the value of data modeling in the Business Analysis stage appears to be undisputed.

There are different approaches to data modeling, as noted by the identification of top-down data modeling and bottom-up data modeling as two separate techniques. However, even within the more widely used top-down technique, there are numerous variants.

Some use constraining relationships, such as the one-to-many relationship, which imply a constraint on the values in one of the entity types in the relationships. Other relationships, such as the many-to-many, are non-constraining. In the interest of recognizing and defining the business rules that the data is required to satisfy, there is merit in emphasizing the importance of constraining relationships.

4.5.6 Suitability of Current Methodologies and Techniques.

There exists an enormous number of methodologies variously referred to as development methodologies and design methodologies. Those carrying the label "development methodology" tend to emanate from an environment that emphasizes procedural programming as an inescapable discipline, and that wishes to systematize this with a view to improving the quality of the systems being designed and built.

Those labeled "design methodology" tend to come from an environment that stresses some kind of business analysis technique as a prerequisite to considering the programming design aspects of the system.

While many early methodologies emphasized one technique, more recently there has been a trend towards "composite" methodologies covering more than one stage and using several techniques.

4.5.7 Future Direction for IRM-Compliant SLCMs.

There are three basic objectives that SLCMs need to satisfy to be compliant with the requirements of information resource management:

1. Cover all stages of the system life-cycle.
2. Provide a choice of techniques.
3. Be computer-aided.

Each of these objectives will be described.

Using the currently available methodologies to accomplish IRM, one uses several SLCMs that must be melded together to produce a single SLCM which will cover all phases from Strategic Systems Planning through the Phase-out/Evolution of a system.

The methodology should provide a choice of techniques. There are many varied techniques to accomplish the different stages of a SLCM. The all-encompassing SLCM should have the flexibility to provide its users their choice of techniques to accomplish that stage of the SLCM. Techniques to be provided are those that the users are more familiar with, feel is better, are forced to use, etc.

A computer-aided methodology will facilitate the SLC. A SLCM with IRM requires a vast amount of data to be recorded, maintained, and analyzed. The inherent nature of IRM requires an understanding of the business process, shared data and information/data analysis, integrated systems development, and an understanding of the inter-relationships which exist are among these factors. This complexity points toward a computer-aided SLCM. This would do away with the need for massive amounts of paper and the situation where only a handful of individuals are knowledgeable about the systems in an organization. The SLCM can be accomplished methodologically, and the users of such SLCM be stepped through it in such a fashion as to give them an understanding of what, why, how, where, who, etc., as each system is being developed.

BIBLIOGRAPHY FOR CHAPTER 4

CURTICE, R. 1986, Strategic Value Analysis, to be published by Prentice-Hall.

DAVIS, G. B. 1982, "Strategies for Information Requirements Determination," IBM Systems Journal, Vol. 21, No. 1, pp. 4-30.

GOLDFINE, A. H., Ed. 1982, Data Base Directions: Information Resource Management—Strategies and

Tools. NBS Special Publication 500-92, National Bureau of Standards, Gaithersburg, MD, September.

GOLDSTEIN, R. C., and McCRIRICK, I. B. 1981, "The Stage Hypothesis and Data Administrations: Some Contradictory Evidence," Proceedings of Second International Conference on Information Systems, Society for Information Management, December, pp. 309-324.

IBM 1981, Business Systems Planning—Information Systems Planning Guide, Application Manual, GE20-0527-3.

KAHN, B. K. 1983, "Some Realities of Data Administration," Communications of the ACM, October, Vol. 26, No. 10, pp. 794-799.

LINDER, J. 1985, "Harnessing Corporate Culture," Computer World, September 23, pp. ID/1-ID/10.

NOLAN, R. L. 1979, "Managing the Crisis in Data Processing," Harvard Business Review, April.

NOLAN, R. L. 1982, Managing the Data Resource Function (Second Edition), West Publishing Company.

OTTEN, G., Ed., 1985, "Information Processing Systems—Database Management Systems—Basic Reference Model, Version 3," ISO/TC 97/SC 21/WG5-5 N232, July, 1985.

PALVIA, P., and NOSEK, J. 1985, "An Investigation into the Usage and Effectiveness of Information System Methodologies, Techniques, and Tools," working paper, Temple University, Philadelphia, PA.

STEVENSON, H. H. 1985, "The Heart of Entrepreneurship," Harvard Business Review, March/April, pp. 85-94.

5. TECHNOLOGIES FOR IRM

Alfonso F. Cardenas

CHAIRMAN

Biographical Sketch

Alfonso F. Cardenas is Professor, Computer Science Department, School of Engineering and Applied Science, UCLA, and consultant in computer science and management, Computomata International Corporation. He obtained the B.S. degree from San Diego State University and the M.S. and Ph.D. degrees in Computer Science, at the University of California, Los Angeles, 1969. He has been a consultant to both users of database technology and to vendors of database management technology. He has led seminars and has spoken at conferences all around the world. He is the current president of the Very Large Data Base Endowment, which conducts the annual International Conference on Very Large Data Bases. He is the author of the book Data Base Management Systems, Allyn and Bacon, Inc., 1979 and 1984; co-editor and co-author of Computer Science, John Wiley & Sons, Inc., 1972; and author of many other publications. He is listed in Who's Who in Computers and Data Processing, Dictionary of International Biography, and American Men of Science. He is a member of the Association for Computing Machinery and the Society for Information Management.

PARTICIPANTS

Christopher Bird	John Mayson
Edward Ferris	Donald McCaffrey
Steven Flanders	Sham Navathe
Stuart Frager	Ronald Shelby
Fred Lochovsky	David Wood
Patrick Mantey	

5.1 INTRODUCTION

The following major technologies are important to IRM:

- o Application Development Methodologies and Supporting Tools
- o Information Resource/Data Dictionary Systems
- o Database Management Systems
- o Application Generation/Development Systems
- o Fourth Generation Inquiry and Report Languages For End-Users (4GL)
- o Systems for New Application Areas, PC/Intelligent Workstations, Local Area Networks and Database Servers
- o Heterogeneous Database Management

These technologies are inter-related, and may make use of other technologies. A number of specific tools with perhaps different names or refinements of the above names are also important and are covered in this chapter.

There is considerable overlap of function between application generators and fourth generation languages. There seems to be no discriminator that would firmly place any such tool into one category or the other. The four major considerations for the purposes of this chapter are:

1. A fourth generation language has features which enable end-users to gain access to their own data without the intervention of data processing staff.
2. A fourth generation language is a true language; it has grammar (e.g., BNF grammar) defining its syntax. An application generator does not have these essential linguistic components.
3. An application generator generates a form of target code which needs further processing (e.g., compilation) before the application can be run.
4. An application generator has the capability of generating not only the application, but also the documentation required to operate the system and the control language for file/database definition and access.

DBMSs, on the other hand, are well delineated by the CODASYL specifications and other de-facto standards such as the relational SQL.

Each type of tool is considered in terms of the following categories:

- o <u>The State of the Art.</u> What is the current technical quality, robustness, and degree of standardization of the tool?
- o <u>The Uses of the Technology.</u> What are the benefits and pitfalls associated with the installation and use of the tool?
- o <u>The Outlook.</u> What are the short-term and long-term outlooks for the tool?

5.2 APPLICATION DEVELOPMENT METHODOLOGIES AND SUPPORTING TOOLS

<u>5.2.1 The State of the Art.</u>

<u>Problem Scope</u>

This section addresses the state of the art in the support of application development methodologies by software. While application development methodologies have been used by application developers for several years, the methodologies are not sufficiently standardized to allow them to be vertically integrated with information planning and database design software tools. While individual software tools are now being marketed that support specific application development methodologies, quantum leaps in productivity will not be seen until application development methodology software is integrated with information planning software and application and database design software. Once this occurs, significant productivity increases will result. Software to support information systems planning activities is expected to emerge soon from its current primitive stages.

Existing Development Methodologies

Structured development methods have all centered around defining requirements and designing and constructing application software and databases to meet the requirements. Structured methodologies came into existence in an effort to ensure that user requirements are clearly identified and met by application software, and that communication problems during the development of application software systems are minimized. These methodologies usually have been advocated and supported by independent consultant firms that often have proprietary rights to the methodologies they support. If followed rigorously, these methodologies are usually successful in developing application software in a manner that leaves a clear audit trail as to the requirements and design of the application.

Unfortunately, however, existing structured methodologies have proven to be slow and labor intensive, and sometimes have resulted in application software that does not meet user needs. The chief problem in the area of speed of application development and labor costs is that most methodologies depend upon pre-defining requirements in textual form. Often a conceptual gap occurs between the users' understanding of the written requirement and the need for automation as the user perceives it in the workplace.

A way of shortening the amount of time required to define specific user requirements has been to prototype an application to define requirements. Prototypes are created quickly using new programming languages or screen painters, thus allowing the end user to view a prototype application system to see if it meets her/his requirements. Development methodologies that use prototyping are very new, and are still evolving. To date, there are no standard methods of prototyping during application development.

Barriers to Methodological Effectiveness

It is our belief that the intuitive approach to system development predominates in the world of application development today. Even though there is widespread discussion of structured and ordered development of application software, the rigor, time, and cost required to follow existing application development methodologies prevent their widespread use. When this is combined with the fact that system development methodologies do not always result in the development of applications which accurately meet user needs, it is easy to see that structured application development methodologies are not yet fully accepted.

A major contributor to the amount of time and cost required to follow development methodologies today is the creation and maintenance of documentation. While most documentation of application development is still done on paper, the automation of documentation has increased. A final barrier to the acceptance of application development methodologies is the resistance of system developers to structure, control, and change the current method of developing systems. Like most practitioners in a "professional" field, application developers desire control over their own work processes and output. This control frequently limits the manner in which an organization can direct system developers to revise their work methods. This resistance to change by system developers is a large barrier to the acceptance of standard development methodologies.

5.2.2 The Uses of the Technology.

Current Application Development Technology

Application development is supported today by three types of technology. The first is computer stored text used to record and recall documentation about an application system under development. This method of storing and retrieving documentation stores information at the document level as text documents. The usefulness of this method of storing documentation for improving the productivity of application development in the future is very limited, since documentation is designed to be used only during the life cycle phase in which it is created. Maintenance of this type of documentation is prohibitively expensive for the benefit received.

A second technological support for application development is the use of an information resource dictionary systems (IRDS) for the storage and retrieval of application documentation. Information in an IRDS is frequently stored as text and numeric data with imbedded cross-references to allow the retrieval of information based on a number of different parameters. This method of storing application documentation offers the best hope for decreasing the cost and time required to develop applications in a structured fashion. Perhaps the greatest cost saving offered by this method of application development support is not during the development of the system itself, but during the maintenance and enhancement of the application. Since maintenance and enhancement is generally estimated to account for 65% - 75% of the total life cycle cost of an application, the economic benefits of pursuing this avenue of technological support is obvious.

A third technological path being followed in supporting application development is not related to application documentation, but to the creation of prototype or "shell" applications using fourth generation languages or screen painters. This use of technology is really an effort to decrease the amount of time and miscommunication that occurs when user requirements are defined. Prototyping tools are likely to speed the user requirements definition process considerably.

Barriers to Effective Technological Support for Application Development Methods

The lack of standard application development methods is the greatest barrier to the development of technology that can decrease the time and cost of application development. In the area of databases the International Organization for Standardization and the American National Standards Institute have defined a standard three-level database architecture which allows vendors to produce technology at one or more of the architectural levels. No such standard currently exists in the area of application development, and until it comes about it seems unlikely that significant progress will be made in creating portable, integrated technology solutions to the application development technology problems of today.

A second major barrier to the use of technology supporting application development methods is the structure of third generation programming languages, such as Cobol and Fortran. These programming languages allow an entire application, both function and data, to be defined by an application programmer. "Business rules" are imbedded in programming code and hence cannot be changed thoroughly or efficiently. By not providing a separate and enforceable division between data and program logic these languages make the enforcement of application development methods difficult, if not impossible.

A third barrier to effective development support by technology is the fact that current technology does not provide automatic and easy access to documentation of the work being done during application development. Once other areas that create barriers are removed, a new technology which results in documentation being created automatically from the development process and the automatic updating of previously documented application development information will need to be created. This documentation should be created and updated automatically to keep development costs down.

A final barrier is the cost of learning curves for staff to master and use new technology. Whether new technology is a programming language, a database system, an application development methodology, or a new technology to support application development, the learning of new work routines is a significant and costly barrier to the acceptance of new technology. The cost effectiveness of new technology to support application development will have to be high to justify the overcoming of these learning curves.

Integration With Other Technologies (Tools)

We believe that the only integrator between information planning tools, database and system design aids, and application development is the IRDS. As the repository for metadata describing functions, data, and technical objects that compose an application, the IRDS is the key integrating tool for application development automation in the future.

The functionality desired for such an IRDS includes the capturing of the general requirements for an application from an information planning tool, and the use of those parameters to generate a prototype of the planned application automatically. Once a prototype is altered to the satisfaction of users, the IRDS documentation should be automatically updated from the prototyping module to provide complete documentation of the system requirements used to create the application system.

A bi-directional information transfer of metadata (documentation) between the application development support tool, the prototyping tool, and the information planning support tool should also be present. This "feedback loop" would assure that when a system was changed during its development or during its enhancement in the maintenance phase of its life, the metadata supporting the information planning tool would also be automatically updated. This would provide multiple levels of integrated documentation within an organization, and allow inquiries about the multiple dimensions of an application system. The result would be a large decrease in the cost of developing and maintaining application systems. It would also result in application software that could be changed quickly.

Naturally, the development of an IRDS as sophisticated as we are describing in this paper will require rigorous design standards for information planning, database and program design, and application development methodologies. Automating the support of application development methodologies is no different than automating other business functions. Before complete automation and vertical integration can be completed, the function must be standardized and fully described. This is not the case with information system planning and application development today.

Only after a higher degree of standardization exists will vendors spend the money required to produce complete and vertical integrated application development support technology. We believe that when this technology is available it will be based around a central repository of metadata—an IRDS—that will radically decrease the cost to develop application systems after an information planning process has occurred.

Section 5.3 provides more information on the IRDS.

5.2.3 The Outlook.

Short-Term

In the next 3 - 5 years we anticipate the continued, uncoordinated development of software tools to support specific application development methodologies. This movement will assist individual application development vendors to decrease the cost of implementing their particular structured application development methods. We see continued attempts to extend the existing programming languages through application generators to speed and reduce costs in application development. While some economic benefits should be derived from this course of action, we do not see it answering the long-term question of decreased time and cost for application development because of the inherent limitations of COBOL and FORTRAN as programming languages, and because application development only accounts for 25% - 35% of the total life cycle cost of an application.

An increasingly important trend in application development is the continued refinement of the prototyping process using 4th generation languages and screen painters. As experience with these methods of defining user requirements continues, we expect some decrease in the cost of creating and maintaining application systems. Application maintenance cost should decrease due to the improved accuracy of meeting user needs in systems developed using this method of defining user requirements.

Long-Term

Over the next 5 - 10 years we foresee an increased trend toward developing the separation of application system levels into something like the 3 level database standards existing today. This will make the support of heterogeneous programming languages and processing environments by a single IRDS-driven application development methodology possible. Applications will be created at the conceptual level and automated code generators will generate the application version needed for a particular operating and processing environment.

In the long run we see the development of the IRDS as the central storage location for all application and database metadata. In this role the IRDS will contain information about the information requirements of the organization, the detailed application specifications utilized to meet those information requirements, and technical system documentation of the software used to run the application. Eventually, application development will be much more automated than is presently the case. Just as more than three decades passed between the invention of the automobile and the creation of a mass production process that greatly lowered the cost of producing an acceptable product, we believe that a similar length of time will have to pass from the beginning of online system development in the 1960s before automated application development is accepted. Today we stand on the threshold of that acceptance.

5.3 INFORMATION RESOURCE DICTIONARY SYSTEMS (IRDS)

5.3.1 Description.

Definitions

a. Information resource dictionary system (IRDS)
   (1) A computer software system that provides facilities for recording, storing, and processing descriptions of an enterprise's significant information resources.

   (2) A computer software system that maintains and manages an information resource dictionary.

b. Information resource dictionary (IRD).
   (1) A collection of entities, relationships, and attributes used by an enterprise to model its information resources environment.

   (2) A repository of data concerning the information resources of an enterprise.

The Role of Data Dictionary/Directory Systems and Database Management Systems in IRM.

The Information Resource Dictionary System evolved from the data element dictionary/directory systems (DED/DS) of a generation ago, to the data dictionary/directory systems (DD/DS) of today, as the need for metadata beyond data identification and specifications became apparent. Many enterprises have adopted commercially available data dictionary/directory systems for information resource management (IRM) purposes. Some have adopted database management systems or file management systems for the same purposes.

The description and assessment of data dictionary systems in the previous Data Base Directions Workshop proceedings still largely holds. Hence, it shall not be repeated here except where necessary.

5.3.2 The State of the Art.

Technical Quality of DD/DS

Since the IRDS standard discussed below has not yet been approved, we must necessarily address the technical quality of the most available tool currently adopted for IRM purposes, i.e., data dictionary/directory systems.

It was the feeling of a number of members of this Working Panel that current data dictionary/directory systems, in general:

- o Lag database management systems in maturity.
- o Provide good support for data management.
- o Do not provide the desired flexibility and range of functions required for the perceived scope of IRM.

Robustness

In support of functions for which they were designed, data dictionary/directory systems are considered to be robust, managing most contingencies. What they do, they do well.

Standards

Current data dictionary/directory systems are different syntactically and semantically, due to the lack of a standard. However, the more robust DD/DS commercially available show similar functional capabilities.

Since each enterprise may define the scope and depth to which it desires to apply the concept of Information Resource Management, such tools as the DD/DS or DBMS may be considered quite adequate. However, many enterprises want an IRDS that is not only designed with their perceived IRM requirements in mind, but which will provide a broader set of capabilities than is supported by current vendor software products.

This need resulted in a movement to develop a proposed voluntary industry standard for an Information Resource Dictionary System (IRDS) to serve both commercial and government agency needs. This culminated in the formation of Accredited Standards Committee X3H4 (ASC X3H4), Information Resource Dictionary Systems (IRDS), in 1980, to develop a standard (specifications) for such a product. It should be noted that the previous Workshop, Data Base Directions III, provided a significant impetus to this development.

A joint draft proposed American National Standard (dpANS)/draft Federal Information Processing Standard (FIPS) has been developed by ASC X3H4 and NBS.

At the time of this writing, the dpANS IRDS consists of:

(1) A Core Standard (Part 1), containing:

a. A standard Minimal IRD schema.

b. An IRD schema extensibility facility.

c. An IRD schema change management facility.

d. An IRDS command language.

e. The semantics for a panel interface.

(2) Several standard optional modules, supporting:

a. A Basic Functional Schema (Part 2).

b. An IRDS security facility (Part 3).

c. An extensible life cycle management facility (Part 4).

d. A macro language facility (Part 5).

e. An application program interface facility (Part 6).

(3) A Technical Report/Guideline, not mandatory, describing IRDS support of standard data models (the NDL and SQL standards).

An illustration of the relationships of the above dpANS, and possible future components of an IRDS, is provided in Figure 5.1, Levels of the dpANS IRDS.

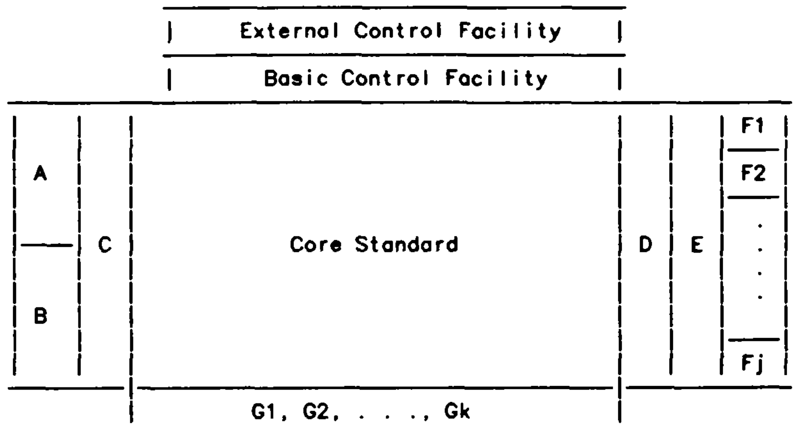

Figure 5.1: Levels of the dpANS IRDS

Figure 5.1 is a conceptual model of the IRDS illustrating how the currently proposed and future components of the IRDS might be viewed as levels or layers surrounding the Core Standard.

The following are merely "suggested" for model visualization purposes:

Core Standard: e.g., minimal schema, schema extensibility, schema change management facility, command language, semantics for panel interface.

Basic control facility: e.g., core security, life-cycle-phasing.

External control facility: e.g., entity-level security, integrated quality indicators/life-cycle-phasing, referential integrity.

A: An extended validation function.

B: A knowledge base interface.

C: A rule-based language.

D: Data types and data-oriented schema descriptors.

E: General external software interface.

F1; F2; ...; Fj: Members of specific external software interfaces set; i.e., COBOL, PL1, Ada data structure generation facilities; SQL/NDL "metadata interface" software; OSI/Local system/application specific directory services.

G1; G2; ...; Gk: "Others". An inner layer (G1) could be a system standard schema; the next layer (G2) could be data management support. These may be either IRDS Modules or Technical Reports/Guidelines.

The dpANS IRDS is currently undergoing U.S. public, Federal, and international review. Anticipating timely resolution of comments from that review, the standard could be promulgated as early as mid-1986.

5.3.3 The Uses of the Technology.

Advantages

The Working Panel felt that the expected ANS IRDS is sufficiently open-ended and flexible so that it can, or has the potential to:

1. Support IRM to any reasonable scope and depth.

2. Interface with, provide metadata to, or exchange metadata with all other information processing tools and technologies.

Pitfalls and Hurdles

The obstacles expected to be encountered by the implementation of an ANS IRDS are generally those already encountered in the use of DD/DS and other tools of IRM. These are categorized as:

1. Human Factors

   a. Unrealistic expectations of users.

   b. A new implementation in support of IRM is construed by many as empire-building.

2. Quality Control Problems

   a. The tools available for mass-loading of previously developed metadata tempts users to load without quality control.

   b. Pressures to load metadata quickly during the several phases of the information system life-cycle to support applications and database design and development, etc., inhibit proper and timely quality checks.

5.3.4 The Outlook.

Short-Term

1. The dpANS IRDS is expected to be accepted by the commercial/industrial and Federal communities with no significant, major change. Some in the information processing field, noting the lack of vendor adherence to past standards, have expressed reservations on the market-place viability of the dpANS IRDS. The participation of a number of vendors in both the ANS TC X3H4 on Information Resource Dictionary Systems and the several National Bureau of Standards sponsored IRDS Vendor Workshops are seen as a positive indication of a more than casual vendor interest.

2. Additional IRDS functions have been identified, and priorities for the development of additional standard modules should be determined in the reasonably near future. Below is a tentative list, not necessarily in the order of priority, of some of the functions that could be developed:

   a. Life-cycle-phase/change control.

   b. Data structure definition/generation.

   c. Support of n-ary relationships.

   d. External interfaces.

   e. Distributed database support.

   f. Evolutionary life-cycle/configuration management support.

   g. A more complete architecture of controls.

Long-Term

We anticipate IRD software that will:

1. Enhance model management, to better handle, for example, graphics/images, voice, and non-traditional data types.

2. Capitalize on and apply to the IRDS, any software which may be developed to integrate text, data, graphics, and voice recognition/speech.

3. Support the development of a standard database access and query language.

4. Pending the development of the above, develop or support development of software suitable for IRDS implementation of IRDS autodial, logon, logoff of internal/external information resource databases available to an enterprise.

5. Lacking a standard inquiry/report language, develop a "standard" IRDS syntax for that purpose and develop modules to map the query to the syntax required of the accessed information resource database.

6. Develop software to provide the necessary foreign language translation of non-numeric query elements and responses to the preceding, where necessary.

7. Capitalize on future enhancements of communication networks to better integrate:

   a. IRDSs in decentralized and distributed environments.

   b. The IRDS interface with personal computers and related local area networks (LANs).

8. Provide enhanced support to other information processing related technologies by the development of additional standard modules to the IRDS.

5.4 DATABASE MANAGEMENT SYSTEMS

A Database Management System (DBMS) is a generalized software system that usually provides a high degree of: data independence, data shareability and minimization of data redundancy, ability to relate data entities (files), integrity, security, performance, and ability to easily access data. Readers are assumed to have basic understanding of DBMS technology.

The use of DBMS has experienced phenomenal growth in the past 15 years. Now, new DBMS are announced frequently, and more types of data models are being defined. Although horror stories concerning implementation of systems using a DBMS abound, the successes far outnumber the failures. Probably the greatest cause of failure is the assumption that installing a DBMS will magically cure all of the problems of Data Management (as well as many other kinds of systems problems). Installing a DBMS does not automatically mean that a "Database Approach" is being taken.

5.4.1 The State of the Art.

The state of the art today is that DBMSs are now seen as a necessity for IRM. Currently, CODASYL and hierarchical systems tend to predominate on mainframes, but relational systems (or relational-like user interfaces) are rapidly evolving. At the micro DBMS level, relational systems now predominate.

There are a number of products available now which claim to be relational; some of these products are often "marketing" or "quasi-" relational. There are many definitions of what makes a product relational. Current thinking considers a relational DBMS as one:

- in which the data may be perceived as being stored as rows in tables with no user-visible navigation elements.
- in which tables are related implicitly (through common fields or attributes) rather than explicitly (through some pointer oriented method).
- able to support directly a relational algebra including, minimally, SELECT, PROJECT, and JOIN.
- able to enforce (at least) the constraints of entity integrity and referential integrity.

As measured against this definition, few commercial products are truly relational. There are, however, many useful "quasi-relational" products that support some aspects of the definition given above. There are also, of course, many DBMS products which are not based on the relational model and these products also do a very satisfactory job of implementation of applications and databases.

Technical Quality

DBMSs are, in many senses, the adolescents of the tools world. They are not yet fully grown and yet exhibit many symptoms of maturity. Each DBMS implements a fairly narrow database mode and must conform fairly closely to the selected model (hierarchic, network, or relational).

Robustness

Robustness here is being taken to mean some measure of "how often do they break," "how often do applications supported by them break," and "how easy is it to change the schema even if nothing has broken" (schema robustness). Most of the DBMSs have been available long enough that the products are fairly robust. Applications supported by the DBMSs are generally robust too, although recovery after failure is difficult with most DBMSs. The ease of recovery does not have anything to do with the underlying model. Schema robustness is altogether a different issue. The older implementations tend not to allow for easy schema modifications while the newer ones do. The relational model encourages "field level access" (and therefore addition of new field types without restructuring) by the "existence" of PROJECT. There is no such impetus with the other models.

Standards

Informal DBMS standards have been around for 15 years (CODASYL 1970, CODASYL 1974). There are standards being proposed for the relational approach (SQL) and there is a new Network Database Language (NDL) that formally standardizes the CODASYL specifications. In the commercial arena, less than 50% of the installed product base conforms to any kind of standard. It seems that for DBMSs, sector is concerned, standardization is both necessary and desirable.

5.4.2 The Uses of the Technology.

There are a number of reasons why the use of the technology of DBMS has had a very positive effect on organizations. There are also some reasons why the adoption of DBMS has had a negative impact.

On the positive side, adoption of DBMS has:

- improved shareability and concurrent access to data. Again, a central data management facility has to be able to allow multiple users through the facility at the same time.
- improved data integrity. Single servers, in control of the locks, can detect simultaneous update attempts, deadly embraces, etc.
- improved programmer productivity due to the above and due to the use of copy book/dictionary methods.
- improved availability and recoverability of systems—centralizing data management leads to centralizing back-ups/recovery, etc.

On the negative side, adoption of DBMS has often:

- implied that DBMS means "database approach."
- raised too high the expectations of end users and DP staff.
- caused resource utilization to be (unreasonably) increased. This is often because of increase in database accessing and types of processing, the handling of concurrency and record locking, stronger security controls, etc.

Some other general negatives are concerned with security and the enforcement of constraints. Many of the constraints in an organization belong in the IRDS and are identified during the development of the business requirements. These constraints need to be enforced by a combination of the IRDS and DBMS. As far as security is concerned, some DBMS do have security to the data field level, but most organizations have disabled it. Many organizations claim to need high levels of security, but frequently do not use the feature when offered. Security profiles should also reside in the IRDS.

5.4.3 The Outlook.

DBMS are here to stay for many years, much like operating systems. In the short-term, it is likely that the following will occur:

- More "pseudo-relational" products will appear on the market.
- User interfaces to existing products will be improved.

- There will be more (and better) integration with the data dictionary/directory system (DD/DSs becoming more "active").
- More application interfaces, e.g., with graphics, will appear.

In the longer term, there are several directions that products might take. Among them are:

- Richer underlying models (semantic models, object oriented models) and new data models for images/pictures, voice, etc.
- Support for rule based systems.
- Greater utilization of (cheaper) resources.

5.5 APPLICATION GENERATION/DEVELOPMENT SYSTEMS

An application generation/development system may be loosely defined as a system striving to develop/generate an application or set of applications while automating many of the tasks of DBMS languages (database definition and database accessing), processing data communications (DC) systems (screen painting and management), and procedural programming languages such as COBOL, etc.

5.5.1 The State of the Art.

We will use the short form "application generators" for application generation/development systems which are now emerging and rapidly evolving. The field is in a state of rapid flux. Every DBMS vendor now offers some kind of an application generator, but each has its own unique syntax and degrees of capabilities.

Technical Quality of the Generators as Tools

The generators are still somewhat immature in that they do not yet provide much integration with existing data dictionary/directory systems and other tools that assist in the analysis and design phases of projects. The generators themselves give little assistance during the analysis phases, but can give considerable help with prototype driven design. The human engineering of the generators is still poor, with very little ability to customize the tools and very limited use of personal computers and graphics. Most of the more encompassing generators have been available on larger computers, but there is now a major emergence of such software for personal computers as well.

Technical Quality of the Generated Code

The code produced by application generators is usually equivalent to that produced by a competent programmer with 2-3 years experience. It is usually not necessary to optimize the emitted code. The code may be portable, being able to run in multiple environments. Most generators produce structured code which is relatively easy to follow.

Robustness of the Tool

Robustness, in this context, means the fit and strength of the tool in the various environments in which it might be used. For application generators there are significant application areas which do not fit well with the tool. There is also a significant cultural barrier preventing the tool from being completely successful. There does need to be a very firm commitment to the tool from the data processing community before its benefits can be fully realized. These tools work best in centralized data management environments, rather than in application specific data environments.

As for limitations, the tools are usually ineffective in handling applications that need real-time (e.g., process control) data acquisition, and systems that require access to data from all over the database, or where the data structures are not managed by the underlying or required DBMS. The functional capabilities of application generators are rapidly evolving.

Robustness of the Generated Applications

Robustness here involves the ability to produce code capable of being restarted, and the ability to regenerate the system to handle changes. In both of these categories, the generators do well.

Standardization of the Tools

There is no immediate strong need or perception of benefit to standardizing application generators. The possibility of a standard being adopted appears remote even if need or benefit becomes strong enough; the reason is the large variety of application generators already commercially available, using the huge variety of non-standardized DBMS and DC systems.

Standardization of Generated Code

Code produced by the generators does conform to standards in those cases where the code is in a programming language for which there is a standard. There is a need to ensure that this situation continues.

5.5.2 The Uses of the Technology.

There are many benefits claimed for application generators. However, as with most of the newer tools, the overriding message seems to be "we haven't realized the potential." Table 5.1 is a very brief synopsis of areas of major benefits and pitfalls.

Benefits	Pitfalls
Improved Productivity	Magnitude of productivity improvements not as great as expected
Building from logical design as an aid to prototyping and portability of generated code	Some percentage of maintenance still performed on generated code
Good time to re-evaluate installation life cycle standards	Generators emerging tied to specific DBMS, DC and related products
Less need to pay attention to technical details of data processing; more attention to business needs	

Table 5.1: Major Benefits and Pitfalls

5.5.3 The Outlook.

The future of application generators seems to be assured. In the short-term there is likely to be:

- o a proliferation of products appearing with concomitant reduction in price.
- o vendors concentrating on narrower application domains and selling special solutions to parts of the overall problem (e.g., screen painters/fast prototypers).
- o some integration of products with existing dictionary and DB/DC systems.
- o for large scale products, more support for analysis and design activities.

For generators to continue to make their mark in the long-term, we shall see:

- o more mature products with improved human engineering.
- o much integration with existing dictionary systems, DB/DC systems and other currently fragmented tools (e.g., database design aids) into an integrated environment.
- o methodology independent products.
- o total coverage of development life cycle, with the emergence of "whole systems generators" or "software factories."

5.6 FOURTH GENERATION LANGUAGES

Fourth generation languages (4GLs) have been with us since 1974 (at least), but it is only recently (the last five years) that they have been classified as a group. Initially, 4GLs consisted primarily of generalized file management systems/sophisticated report writers. As with DBMSs, the growth of 4GLs has been very rapid. Many of the 4GLs are moving towards becoming full function DBMSs, but they sometimes lack the data management power to do this effectively. It is often confusing and not clear cut where the actual functional turf boundaries are between 4GLs and DBMSs and application generation/development systems.

5.6.1 The State of the Art.

4GLs are often thought of as "state of the art" tools and yet, while the notion certainly is, the implementation all too frequently is not. The technical quality is often uneven, the products lack orthogonality, they lack recursive facilities (making formal specification very difficult) and they often suffer from severe run-time performance problems. The lack of orthogonality may be a contributing factor to the performance issue, since it may not be possible to describe the syntax of some 4GLs in a formal specification language, and may prevent the writing of compilers for them.

The products are generally robust in that they frequently meet end-user needs and are not easy to "break."

Within the topic of standardization, there is probably a need for a core standard so that some primitives can be defined across the board. It is probably too late to produce a standard which would be acceptable to commercial vendors and/or the commercial client base. For an organization using a 4GL, there is the practical need for a set of standards/guidelines specifying when to use the 4GL and when not to. The boundary between when to and when not to may well disappear as 4GLs become less resource intensive and the various types of tools merge/integrate more.

5.6.2 The Uses of the Technology.

Once again the negative side of the tool centers around excessive expectation, frequently as a result of overzealous marketing. There are, however, a number of pitfalls associated with the use of 4GLs.

- o using a 4GL for the wrong kind of application.
- o 4GLs (by virtue of their design) tend to be resource hogs.
- o for end-users, 4GLs becomes more difficult to use as the application becomes more sophisticated.
- o it is often difficult (particularly with SQL-like languages using non-simple constructs) to understand first how the 4GL understood the query and then how it was implemented.

Some of the major benefits realized using a 4GL are:

- o very fast development for both "quick and dirty" and regularly scheduled programs.
- o may be the most convenient way of accessing data controlled by many different data management environments, DBMS and non-DBMS.
- o ease of prototyping. Prototypes can be built quickly; however, once the prototype has been constructed using the 4GL, rebuilding the system using "production methods" may be time consuming since very little can be salvaged.

5.6.3 The Outlook.

In the short-term, the trend towards developing full function DBMSs from the 4GLs will continue. The 4GLs themselves will become more and more like natural languages.

The other major short-term direction to be taken will be in the area of performance improvements. Translators/ compilers are already under construction; other approaches to allow a speeded up "run from the source" will also be tried.

In the long-term, 4GLs and application generators will co-exist, possibly with a new set of 4GLs which can feed off the production dictionaries maintained for/by the generator. It is certainly possible that the 4GLs will be replaced by the combination generator/query language approach.

The next (fifth generation) software toward the 1990s will most likely involve a much better integration of the various types of software tools: IRDS, Application Generators, DBMSs, 4GLs and DC systems. The multiplicity of current languages and software module flavors will fold into, it is hoped, a smaller and more manageable number under one environment.

5.7 NEW APPLICATION AREAS AND PC/INTELLIGENT WORKSTATIONS, LANs, AND DATABASE SERVERS

New applications and advances in technologies are mutually moving information resource issues into new arenas. Applications such as computer-aided design, engineering and publishing, office and manufacturing automation, and decision support all have significant information resource management requirements. These requirements translate into quite different specifications for the database systems to support them, as well as different hardware and system environments. Traditional database systems have grown up to serve the requirements of airline reservation systems, banking, order entry, inventory control and finance, and other well known applications. For these applications, particularly banking and airline reservations, the database system must support many simultaneous users, with each user's transaction involving a few records of the database, and for a very short time. The records involved have an inherently consistent structure, and the design of the system is to accommodate frequent updates. The requirements for database support for the new applications are significantly different:

- high volume of data per transaction—typically in applications dealing with images/pictures; the amount of data involved in individual transactions is of the order of hundreds of thousands of bytes coming from a variety of entities/objects.

- high number of data types—in a conventional DBMS, the basic data types mostly present in programming languages are sufficient (e.g., integer, real, date, money, and character string). However, in mechanical CAD applications using complex part geometries, the primitive types may include polygons, surfaces, lines, points, etc. VLSI design applications also deal with 3-D geometries. In a statistical application, multi-dimensional matrices, vectors, time series, etc., may constitute meaningful data types.

- ability to provide multiple perspectives on the same data—information such as digitized scanned images or maps have to be interpreted under various perspectives—as grids, as superimposed polygrams, etc.

- span of an update—in conventional DBMSs, an update applies to a view of data which typically comes from a few objects (relations). The update is carried out if it is unambiguous and its side effects are fully specified. For example, if changing a person's grade implies his salary must also change, that must be pre-specified as a side effect of the update. In applications involving "complex objects," or a cluster of different objects, a total specification of all the ripple effects of an update is extremely difficult to specify under all situations.

- versioning and tracking requirements—versions of data are important in design databases; software project management involves recording data about versions of programs. Applications in medicine, etc., need to track information—such as patient histories—over time. This involves incorporating time as an essential part of the data model.

- mixing of multi-media information—with the integration of technologies, voice and image data may be combined together with textual descriptor information in office systems. In publishing and printing businesses it is very common to combine information of mixed nature.

- multiple sources of data—sometimes data from different sources is combined or interleaved to be stored as a single database. Various transformations, interpolations, and extrapolations become necessary. Allowances must be made for missing data, incomplete data and overlapping data. This situation is typically encountered in experimental observations, geographical or environmental statistics, etc.

- combinations of the above—which are increasingly appearing. In a design or decision support environment, there are few users sharing the same data, and their "transactions" can be measured in hours or days. Many records make up such a transaction, and update occurs infrequently. Records are of various lengths, versions are important, the system must support structured relationships, and the data will be of mixed types including text, graphics, and long fields that will contain images, measurements, etc.

5.7.1 The State of the Art.

The new application areas mentioned above are still evolving in their use of computer technology. Many of the application areas bring together, in new ways, technologies such as software, workstations, and networks. In many cases, the applications are pushing the frontiers of the ability of the technology to deal with them effectively.

Often, the systems that address the needs of a particular application area are stand-alone, and do not interface easily with other information processing systems within the organization. In addition, the quality of the systems that are available today is uneven, with many vendors vying for a share of the market. In such a situation there are many systems to choose from, with few criteria as to how to make the choice. The potential buyer is often easily bewildered by the claims made about a product and unsure how to choose the right system.

Because of the evolving nature of these application areas and the data representation and processing techniques that they employ, it is not yet appropriate to introduce standards for IRM. However, standardization of data representation and processing techniques that have proven useful may be appropriate.

5.7.2 The Uses of the Technology.

As far as organizations are concerned, the benefit of these new technologies is a perceived increase in the productivity in the application areas that the technology addresses. That is, organizations are able to accomplish more with little or no increase in the number of people employed. For example, CAD/CAM systems allow organizations to design new products much more quickly than by traditional methods. Another way in which these systems benefit organizations is to increase the quality of the products that they produce. For example, office automation systems may allow better quality documents to be produced, which may be a competitive advantage.

The technologies making these new design applications more economical include, in particular, continued reductions in semiconductor memory prices and improved price/performance for microcomputers. Hardware and system manufacturers have translated these technological developments into powerful and affordable personal computer workstations. These workstations are assuming a major role in the support of interactive information processing in applications such as design and decision support, and their growth is expected to continue unabated. It is a reasonable expectation that in the next ten years, most of the user interaction with the information resources of an organization will be via an intelligent computer workstation. This growth in personal workstation power has presented additional problems in the management of the information resource. Now, many users keep their own copies of portions of the information resource in the workstation, where no other management is provided, and other parts of the information resource may never leave the workstation and become part of the collective resource of the organization. Merging updated individual copies back into the collective pool is an extremely complex process. Problems of data integrity, data security, and data sharing are exacerbated by this proliferation of powerful workstations.

5.7.3 The Outlook.

Short-Term

The solution to these problems lies in further advances in information system architecture. This includes both the development of database systems for distributed environments, and the extension of the system architecture to support networks of workstations sharing a central database server. Local area networks (LANs) will provide the connection required between workstations and the central server, but today (and in the immediately foreseeable future) these do not provide adequate data transfer rates to provide a viable alternative to local disk storage at the workstation. This is especially true in the applications where full page raster images are among the records stored in the database and which the user wishes to browse interactively. Bit-mapped color displays of full page size (1-2 M pixels) are fast becoming a standard, and these permit work with full page image documents. LANs would require real transfer rates of about one-hundred times that of Ethernet to service these user requirements. Local workstation storage, including semiconductor memory, magnetic disks, and (soon) optical disks, provides very economical local workstation storage, supporting a sizable local database. In the next few years, it will be common for a workstation to have upwards of 10 M bytes of main storage, 50 to 100 M bytes of disk storage, and the optical disk will make possible access to as many gigabytes of data as the user wishes to provide for; this latter will most likely be in read-only form. Thus, major portions of the information resource of interest to the user will be at the workstation.

In applications like decision support, and many in computer-aided design, engineering and publishing, and in manufacturing automation, much of the data required by the user is only for reference, and is not altered by the user nor modified frequently by other sources. This data will be a natural match for distribution on optical (read-only) disks, with modifications broadcast and stored on magnetic media until it is economical to produce a new version of the optical disk. Unfortunately, there are no database systems today that can simultaneously manage the shared (centralized) data and the data in the individual workstation. It is expected that local storage will continue to be much more affordable than LAN bandwidth, so this problem will not be eliminated by LAN technology.

Database support across shared servers and advanced workstations is not made any easier by the operating system environments, as those of the workstations and servers are rarely the same. (Some UNIX implementations today run on both personal workstations and shared minicomputer servers, and IBM's VM/CMS will run on both the host and the PC/370 workstations—however in both cases these are not completely satisfactory in that full transparency has not been achieved). Shared operating systems between workstations and servers is expected to become more common in the near future. Standards in operating systems, and in LANs, may appear attractive for the user, but in both areas premature attempts at standards do not appear to be appropriate, as the technology is still developing.

In LANs, the promise of optical fibers has yet to see any widespread realization, but should provide significantly greater bandwidth than today's LAN. A major requirement today is for bridge hardware to simplify the interconnection of different LANs. Software to support the movement of data files between different operating system environments is also required. From this discussion, it is observed that the LAN environment of a sizable organization will require administration. This function is closely related to that served by the database administrator, and could quite naturally be merged with that function.

Even with major improvements in LAN data transfer rates, many applications (especially involving significant interactive graphics as in CAD) will require local data copies. Similarly, highly interactive applications with image data will still find local storage or buffering required for satisfactory performance. These applications need to be addressed in terms analogous to a storage hierarchy, with staging/paging strategies developed to achieve economical trade-offs between LAN data rate and local storage cost for a given performance. Some applications may have sufficient locality of reference to permit "anticipatory paging," while others may be better served by a data staging approach which provides to the user the entire context (e.g., an insurance file to a claims adjustor) and then alerts the workstation user that the file is now local and ready for processing.

The new application requirements lead one to believe that the conventional data models—network, hierarchical, and relational are inadequate to deal with the above problems. Although they represent good, generalized solutions to database modeling, they lack two things:

1. a facility for incorporating the semantics of specialized application domains in terms of data types and high level structuring primitives.

2. ability to perform high level operations meaningful under these specialized domains.

Extensions to existing data models, particularly the relational model, may be forthcoming, and are already under consideration. They incorporate ideas such as the use of complex objects and long records to accommodate the demands of the design environment.

The central database management system must, as a server in a distributed workstation environment, provide some new functions. One of these could be called a "subscription extraction" service. In applications such as decision support, the user wants to maintain a personal "snapshot" or extracted set of data for use with models, etc. However, the user would like to have, periodically, a new extraction of the same data items. A subscription extraction would provide this required data, which the user and the workstation database would treat as a version. That is, the user would in many applications want to keep the time sequence of these "snapshots" in the local database. In addition, the central database server could provide the data to the workstation in the format required as input for different applications; e.g., as the input to Lotus 1-2-3. The server would also make possible alerts to users, as when data provided to them in read-only form is being updated by another user on the system. In addition, it could manage the backup of user data, the provision of a common data dictionary, and thereby enable sharing of "personally derived" data among users who wish to allow this to happen. The data dictionary requirement in some applications may become so significant that it becomes (either virtually or really) a separate server.

Long-Term

We foresee that on a more long-term basis, there is a need to continue work in areas such as object-oriented data models and DBMSs. These data models or DBMSs must have the following characteristics to be really useful:

- they must allow users to define their own application domain-oriented (abstract) data types.
- they must provide for a facility to transform data among these types.
- DBMSs must extend the concept of program-data independence to program-data-media independence. By that we imply that the structure and operation primitives should apply uniformly to data residing on different media or derived from different sources (image data, voice data, graphic data, text).

Facilities of the following nature, which are presently not available in DBMSs, need to be provided/enhanced.

- extracting—selective extraction of information (possibly automatically) based on predefined user profiles.
- indexing—allowing multilevel indexing capabilities, especially for mixed media information, maps, etc. (e.g., consider answering queries about maps that involve countries and water management districts, and which relate to some water quality statistics based on river and lake samples).
- *inferencing and reasoning*—*a considerable interest exists in making DBMSs more intelligent in* this sense. Decision support systems, expert systems involving a large collection of facts and rules, need an underlying DBMS. The DBMSs in turn can be enhanced by providing an intelligent interface to support capabilities for ad hoc question/answering, searches based on heuristics and recursion, etc.
- alerting users, triggering updates—better facilities are needed to make DBMSs more active so that they can alert the right users upon the arrival of certain information. Moreover, techniques to cause a controlled propagation of updates need to be provided.

5.8 HETEROGENEOUS DATABASE MANAGEMENT

Because of the proliferation of databases and the diversity of data models and database management systems, an increasing number of organizations now own data stored in a heterogeneous environment. To look upon this as a central resource, two possible approaches can be taken.

1. making a distributed database with geographically dispersed, locally autonomous databases
2. constructing an integrated database by merging the data.

In either approach, the following problems must be dealt with:

- providing a layer of schemas so that external and conceptual schemas are defined both at global and local levels; providing transformations among these schemas.
- mapping data from existing models into some common data model.
- mapping global queries expressed against the global data model into queries against individual databases.

5.8.1 The State of the Art.

No commercial systems are available to deal with data stored in such a heterogeneous environment. However, there are several projects underway in industry and universities to support heterogeneous database schema integration and query processing. The various projects stress a multi-layered approach, from the user level and view, through various internal layers, to the local databases. The internal model and language differ among the projects, although the tendency is toward extensions of the entity-relationship approach and related language with sufficient semantics to capture the essence of the participating heterogeneous DBMS.

In all of the above approaches, there is an assumption that the local and global database schema information is available "somewhere." The ideal place to keep that information is an IRDS. An IRDS for the above context must provide for

- mapping information related to data and queries.
- creation of inter-database relationships.
- specification of inter-database constraints.

Additional problems regarding dictionary placement and distribution of the schema information must be dealt with.

5.8.2 The Outlook.

The user experience with the above approaches is non-existent at this time since the systems and the tools mentioned are not in any usable form. A large number of difficult problems dealing with updates have not even been fully understood theoretically. Dictionary systems will play an important role in the above approach, and much more research needs to be directed towards coping with heterogeneous databases in years to come. The challenge increases in the case of heterogeneous DBMS plus heterogeneous data types (conventional data, pictures/images, text, voice)—a case already faced by a number of organizations.

6. IRM IN A DECENTRALIZED/DISTRIBUTED ENVIRONMENT

Olin Bray

Chairman

Biographical Sketch

Olin H. Bray is president of BIE Corporation, an information systems consulting firm specializing in CAD/CAM data management. He has consulted on the role of data management technology in CAD/CAM and the functional and informational requirements for many CAD/CAM functions in both the mechanical and VLSI areas. He has managed a geometric database design project. His recent work has involved identifying long term and short term data management requirements for CAD/CAM and engineering in both mainframe and workstation environments. He previously worked for Control Data in data management and CAD/CAM, and for Sperry in distributed database requirements and database computer architecture. He has also been the MIS manager for a large health center. Mr. Bray has been a member of the CODASYL Systems Committee and the IGES Steering Committee, and has been an ACM National Lecturer. He has given many conference papers, written two books: Distributed Database Management Systems, and Data Base Computers, and is working on a third: Integrated CAD/CAM and the Factory of the Future.

PARTICIPANTS

Gordon C. Everest	Alexander MacEachern
Elizabeth Fong	Louis I. Reich
Warren Hoffman	Eileen M. Trauth
Martin Kaplan	Bruce J. Vanek
Willard Lackie	Ron Weeks

6.1 INTRODUCTION

This chapter discusses Information Resource Management in a decentralized and a distributed environment. A decentralized or distributed environment requires extensions to IRM as developed for a centralized environment. These changes may be introduced into an organization in either of two ways:

1. As an organization with IRM moves from a centralized to a distributed environment.

2. As an organization with a distributed environment but no IRM begins to introduce IRM.

This chapter discusses both of these types of changes.

Chapter 6 is organized into seven sections plus this Introduction. Section 6.2 provides a framework and sets the scope of the chapter by defining IRM and distributed processing. Section 6.3 identifies the factors that encourage and inhibit the shift toward distributed processing. Section 6.4 describes "spheres of control," a concept for relating organizational factors to ways of sharing data and control among various sites and organizational units. Since an organization's starting point affects how it approachs distributed processing, sections 6.5 and 6.6 describe the two major starting points and the transition planning to move an organization into a distributed environment. Section 6.7 reviews the technology involved in distributed processing and distributed database management, describing the state of the art and trends. Section 6.8 is a summary.

6.2 FRAMEWORK FOR IRM IN A DISTRIBUTED ENVIRONMENT

Both IRM and distributed processing are broad, frequently ill-defined, areas. Therefore, this section provides a framework for the chapter by defining and characterizing these two areas.

6.2.1 Information Resource Management.

Since other chapters in the report have discussed IRM in detail, this section simply summarizes its key aspects from our perspective.

The scope of IRM practiced by an organization should be defined by the information needs of the business. Its scope is not defined by any technical configuration or the scope of individual data processing organizations. For example, if the business needs require interaction outside the company with vendors or customers, IRM must consider ANSI and ISO standards. At the other extreme, IRM need not encompass data that is not relevant to organizations' business needs. This would then exclude working papers and data of interest solely to an individual.

Traditional IRM

Information Resource Management has traditionally involved five basic concepts. First, information is a resource to be managed. Like other types of resources, information has characteristics such as value, cost, quality, and timeliness. However, unlike other resources, information can be shared and used without being depleted or worn out. Organizations are now beginning to recognize the importance of this resource and manage it more effectively.

Second, since it makes no sense to collect and store common data redundantly, information should be widely shared across applications. One of the key functions of IRM and strategic data planning is to identify these common data and manage them appropriately for all of the applications that need them.

Third, because of the importance and value of information, data quality is important. This quality involves data integrity, consistency, and backup and recovery facilities. Related issues also involve security and privacy.

Fourth, IRM implies that the management of the data is independent of the applications using the data. Data models and database designs should be built to model the actual relationships in the "real world," not simply those relationships needed to support the current set of applications. Also, the DBMS, the software operating directly on the data and managing it, should be independent and separate from the application programs which use the data.

Finally, an important goal is to integrate the various applications through their use of common data. It is no longer adequate to allow each application to define and use its own input and output data formats and integrity constraints independent of all of the other related applications.

Extensions to IRM

This section describes five key extensions to the basic IRM concepts described above. Some of these extensions are important even in a centralized environment, but others involve only a distributed environment.

Although in its broadest sense IRM includes both computerized and non-computerized data, its focus has been more on the well-structured data, as opposed to unstructured data, primarily as a result of its outgrowth from the database management area. IRM must be extended to include many types of unstructured data, such as text files produced by word processors and electronic mail systems, and digitized voice and images.

A second extension involves artificial intelligence and knowledge bases. IRM today captures metadata in the form of data dictionaries and schemas. However, for AI IRM must include more information about the meaning of the data, such as more complex integrity constraints, inference rules, and inheritance structures.

The third extension involves the use of external data. Traditionally, IRM has focused on the effective management and sharing of an organization's internal, operational data. However, today organizations increasingly need additional data from external sources. Examples of these external data include economic indicators, census data, marketing studies, and mailing lists. In some cases, such as marketing studies, these data are obtained once for special analyses, but in other cases, as with economic indicators, they are routinely updated and maintained by an outside organization. An organization may query these external databases as needed or periodically copy selected portions of them into its own internal databases for later processing. The important issue is that information and policies about these external data sources and the standards for communicating and converting these external data must be included within the scope of the organization's information resource management.

The three extensions above apply regardless of whether the environment is centralized or distributed. The following two extensions are much more significant in a distributed environment. First, in a distributed environment, IRM must include both computer and communications resources. The manner of data distribution and the policies for sharing them are significant IRM issues in a distributed environment. The communications system's structure and bandwidth is a major determinant of feasible distribution alternatives.

Second, although control and coordination is an IRM issue in a centralized environment, it becomes much more important and complex in a distributed environment.

6.2.2 Distributed Processing.

What is Distributed?

This subsection identifies the types of objects that can be distributed. Distributed processing always seems to include distributed hardware and, frequently, distributed data. Because the distributed environment is the focus of this chapter, this subsection more precisely identifies what is distributed and the characteristics of distributed. There are four types of objects that can be distributed: technology, data, policy, and functions.

The technology objects that can be distributed include both hardware and software. The hardware consists of processors, storage, I/O devices, and communications facilities. The software includes both system software and applications programs. From the organization's perspective it may be the business functions, such as accounts receivable or inventory control, that are being distributed, but from the information systems perspective it is the software, the actual code, that is being distributed. In some respects, software simply represents another type of data to be distributed.

The second type of object for distribution is data. Furthermore, there are three distinct types of data that should be distinguished. First, there is traditional, well-structured data that is stored in data files or most databases. Second, there is metadata, i.e., data describing the data being stored. Third, there is non-traditional or unstructured data such as text, images, and digitized voice. These unstructured data are becoming much more important because of their role in both office automation and CAD/CAM, two rapidly growing areas, both of which are dominated by workstations in a distributed environment.

The third type of object involves policy. These policies may involve planning, development, the ownership of and responsibility for data, and the coordination and control of data as it is defined or moves through the organization.

The fourth type of object is function. Depending on how and what an organization chooses to distribute, the necessary support staff may also have to be distributed. However, in reality it is not the people but rather the functions that are being distributed. From the organization's overall perspective, it is the business functions that are being distributed. However, from the IRM perspective the concern is with the various information systems functions such as analysis, design, programming, training, testing, and operations.

Characteristics of Distribution

Given that one or more of the above types of objects can be distributed, this subsection identifies some of the key characteristics of a distributed system. First, there are three distinct types of systems—centralized, decentralized, and distributed. A centralized system has everything organized, controlled, and performed from one location. A decentralized system has multiple independent locations with essentially no communications between them. A distributed system implies communications and coordination. A distributed system usually involves multiple locations with various types of communications and control among the various locations. The focus of this chapter is on the distributed alternative. The decentralized approach is important primarily because it is one of the two starting points from which an organization begins its migration to a distributed system.

An additional complication is that people tend to think of centralized and distributed as a pure dichotomy—you are either centralized or distributed. However, in reality there are many variations between these two extremes. From the IRM perspective some types of objects may be centralized, while other types may be distributed. For example, the hardware and software may be distributed, while the data, policies, and development functions may be centralized. In other cases the hardware and data may be distributed, while the metadata and its control may be centralized. Therefore, to be precise, a system is distributed with respect to the types of objects that are distributed.

With respect to IRM in a distributed environment, there are two key characteristics of distribution. First, there are multiple copies of one or more types of objects, including at least one technology and/or data object. Second, there is the need for coordination. Distribution may also involve local autonomy and geographic separation of the objects. Distributed processing involves multiple occurrences of hardware, which may be at the same or different locations and which may be the either homogeneous or heterogeneous. This can be done, and in most cases is done today, without a distributed database. Distributed database management requires the further distribution of the data and its management by one or more DBMSs, but without requiring the user or application programmer to be actively involved in locating the data and insuring its integrity.

The other key characteristic of distributed processing involves the need for coordination. This may involve specifying common hardware or simply ensuring a common communication standard so that different types of hardware can communicate. For data objects it may involve ensuring that the data conform to certain common data definition standards and documentation and that the common definition includes all of the integrity constraints needed by all of the user at all of the various sites. It may also involve defining standard procedures for updating the data or its definition. Similarly, policies about access control, security, and backup and recovery must be coordinated.

Two additional characteristics are usually present, but are not necessary for a distributed system. First, there may or may not be local autonomy. Frequently there is much local autonomy subject only to the restrictions incurred for coordination. For example, a site may be able to select whatever hardware and software it wants as long as they support a specified communications interface.

The other frequent characteristic is geographic distribution. Originally, distributed systems were scattered over a wide geographic area, nationally or multinationally. This type of distribution *required certain types of communications systems and placed technical constraints on how certain* technical problems could be solved. Today, two additional types of distribution are possible. Local area networks (LANs) allow distribution over a much smaller area, such as a single building or a small complex of buildings. The second approach involves distributing a system or a database over several virtual machines which may actually be implemented on a single computer. All of these types of distributed systems must perform the same logical functions, such as error detection and correction, concurrency, maintaining consistency if there are multiple copies of the data, and locating and moving data or processes so that a request can be processed. The key difference with these types of distributed systems lies in their performance, especially in the communications area. These performance differences encourage or prohibit certain approaches and implementations for solving the common logical functions.

Domains of Distribution

Considering the four types of objects that can be distributed, there are many different approaches or domains of distribution. Figure 6.1 shows four common domains based on looking at just the technical (the combination of technology and data) and policy areas. Each area can be either centralized or distributed. We could further break down each type into its specific objects.

		Technology	
		Centralized	Distributed
Policy	Centralized		
	Distributed		

Figure 6.1: Domains of Distribution

The focus of this chapter is on distributed technology (where technology includes both hardware and data) and distributed policy.

6.3 FACTORS AFFECTING THE TREND TO DISTRIBUTION

There are several factors and trends either encouraging or inhibiting an organization's shift toward a distributed system. Depending on the specific organization and application, these factors have different weights. This subsection identifies and discusses each of these factors.

6.3.1 Factors Encouraging Distribution.

The factors encouraging the trend toward distributed processing and distributed databases include:

- Reduced processing costs.
- Expensive communications.
- Faster access to local data.
- Higher people costs.
- Increased reliability.
- Increased security.
- Flexible growth/expansion.
- More user control and local autonomy.
- Need to coordinate decentralized autonomous sites.
- Limited hardware/software functionality.

Several of these factors involve a cost/performance trade-off between processing and storage versus communications technologies. Unit costs for processing and storage are declining much more rapidly than for communications. Therefore, whenever possible, an organization prefers to trade communications for processing, i.e., distributing processing power and data to remote sites to reduce the amount of communications. When there is locality of reference this also provides faster access to data because local data can usually be accessed much faster. Therefore, distributed processing is clearly encouraged by the relative costs and performance between processing and communications.

There is also a trade-off between relatively cheap hardware and processing versus increasingly more expensive personnel costs.

The need for reliability also encourages distributed processing. As organizations store more of their data in the computer and use on-line systems to support their operations, reliability becomes critical. Failure of the computer is no longer simply an inconvenience, like a delayed report, it can literally stop the company's operations. Therefore, reliability, especially in the sense of fault tolerant operations, is critical. Loss of a node or a communications link in a well designed distributed system may degrade the system's performance, but it is not a major catastrophe like losing a centralized system. However, redundance of data as well as hardware is essential for this improved reliability.

Security is another factor that becomes important as more of the organization's data are computerized. Distributing the hardware and data resources can increase security through physical separation. Different security procedures and access controls can be used at different sites. However, the increased communications in a distributed system increases another type of vulnerability. Therefore, although a distributed system is not necessarily more secure, it does allow more rigorous security controls if the organization chooses to use them.

Distributed systems also allow more flexible growth and expansion. Adding another node to a network is much easier than upgrading to a different, more powerful computer family when you run out of processing power or storage capacity.

The demand for more user control and local autonomy is also driving organizations to distributed systems. This trend has been set by the decline in processing costs, the ease with which these systems (especially microcomputers) can be used, and the application development backlog and apparent unresponsiveness of corporate data processing departments. Many users, rightly or wrongly, feel that they can do a better job satisfying their information system requirements than a centralized data processing department. Opposing this trend is the fear, primarily by information systems professionals, that the organization will lose control of its data resources.

All of the above factors encourage the movement toward distribution from a centralized environment. However, today many large organizations have a decentralized rather than a centralized information systems environment. They have multiple, relatively independent data centers often with different, even incompatible, hardware and software. The need to coordinate the activities at these independent sites and to make better use of all of the organization's information resources is a strong incentive for these organizations to begin to link these centers and move from a decentralized into a more distributed environment.

A final factor encouraging distributed systems, specifically heterogeneous system, is the different capabilities of different types of systems. Specific systems are better for time sharing, transaction processing, database management, rapid prototyping, or other functions. Rather than having a homogeneous system, large organizations may choose several different systems, each optimized for a particular function. For example, they may have a large mainframe for the corporate database, extract data and download them to microcomputers for a spreadsheet or other type of analysis. These heterogeneous distributed systems require sophisticated communications systems and facilities which hide most of the differences from the users.

6.3.2 Factors Inhibiting Distribution.

The factors inhibiting or slowing the movement to distributed systems include:

- Centralized management philosophy.
- High past and existing investment in current systems.
- High conversion costs.
- Fear that MIS will lose control.
- Lack of adequate hardware/software tools and technology.

One major inhibitor of distributed systems is a widely entrenched centralized management philosophy. However, in all organizations, top management always wants and needs some central control and coordination (such as consolidated financial statements), while users generally want more local control and autonomy. This centralized approach affects both the general management of the organization as well as the information systems management area. It is difficult for one part of an organization or one corporate function to have a different management philosophy than the rest of the organization. The resistance of many corporate MIS departments to widespread end user computing and microcomputers is just one indication of this centralized philosophy.

A second inhibitor is the large investment many organizations have in their existing centralized information systems. Many of these systems support large, complex operations and companies are hesitant to redesign and rebuild these systems, especially when there is a backlog of unmet needs. This may become less of a factor as companies focus more on the competitive advantages provided by more sophisticated, state-of-the-art information systems and less on minimizing information systems expenses.

A related inhibitor is the high conversion costs involved in shifting from either a centralized or a decentralized to a distributed system. This may involve retraining of both information systems and user personnel and the development of, conversion to, and enforcement of new standards.

Another inhibitor is the fear that centralized corporate level MIS will lose control. There are two ways to interpret this fear. First, there is frequently the fear by the MIS department that it will lose its influence and control, i.e., that information systems dollars and personnel will be shifted into various functional areas rather than being centralized. In some organizations this fear is very real and must be dealt with. However, there is a potentially much more serious problem. This is the fear that not just MIS but rather the entire organization will lose control of its data resources. It is the fear among competent data administrators that if the organization distributes too much data and control too quickly without adequate controls, no one will be in control. Many organizations have worked long and hard to build up data administration and database administration functions. The real fear is that we simply do not yet know how to adequately control and administer data in a distributed environment.

A final inhibitor is the lack of tools to help design, build, and maintain distributed systems. Virtually all of today's application development and database design tools are for a centralized environment. In the past, a few large organizations built special purpose, customized distributed systems. However, for distributed systems to become reasonable choices for many organizations there must be general purpose tools such as design aids, distributed operating systems, and distributed database management systems on which they can be built.

6.3.3 User Requirements.

Organizations are migrating to systems and procedures for managing data and applications to optimize their overall organizational objectives. Trends in hardware, software, and communications will allow placement of data and applications at the "best" levels and sites. IRM procedures must mature to support this view.

The typical organization has, or will have, centralized, decentralized, and distributed levels and sites. Data and applications at each level will be determined by business needs, corporate culture, and even arbitrary criteria. In addition, most organizations will communicate with external organizations including customers, vendors, financial institutions, and government bodies.

Specific requirements for IRM to support decentralized and distributed environments include sharing data among all decentralized/distributed levels and sites and defining and enforcing standards.

Implementation of true distributed environments will require both easy movement of data from site to site and accessing of single collections of data by applications running at multiple sites. Ideally, the former implies a distributed database management system and the latter implies a distributed operating system.

As organizational units must increasingly communicate with each other and with external organizations, communications and data definition standards will become an important part of IRM. This will involve incorporating a hierarchy of standards including international (ISO), national (ANSI), industry, and corporate standards into internal IRM procedures.

IRM procedures which support decentralized/distributed environments must be implemented so that the individual sites do not incur a large administrative burden. Since each site is an organizational unit with its own goals and objectives, IRM must not inhibit it from meeting its own business objectives.

6.4 SPHERES OF CONTROL

This section discusses the ways in which data are used and controlled in a distributed environment. There are three ways in which the data can be used—locally, interchanged, or shared. Control of the data may be local or shared.

For this discussion, data or policy objects are distributed to an organizational unit.

6.4.1 Local Data.

Local data originate and are used exclusive by a single organizational unit. Other parts of the organization do not need to know how these data are defined, collected, validated, or stored. Local data may also include external data, such as market research data, if they are used only by a single unit. In some cases these local data involve only local operations so they are of no interest to the rest of the organization. For example, detailed operational data for a unit is rarely needed by higher units, although they may get summaries of these operational data. In many cases, however, these data are processed to provide information that other parts of the organization need. This information is then shared with the rest of the organization via interchanging or sharing.

6.4.2 Interchange Data.

From the technical information systems perspective, interchanging data is the simpler form of data sharing between two organizational units. However, from a user perspective it may be the more complex because there is usually no transparency in the process. A copy of the data is physically transferred between the two units. Depending on the specific arrangement, the data may be formatted as the originating unit produced it, as the receiving unit needs it, or in some standard exchange format. A common exchange format is more common when the data are used for many purposes by several units. An important point about interchanged data is that they represent a snapshot of the data when they are exchanged. Interchanged data are transferred on demand or on a fixed schedule, but once a copy of the data is passed to another unit, the relationship among the copies is lost. Changes to the data in the originating unit are not automatically passed on the receiving units to update their copy of the data.

From the computer and software perspective, this interchange of data is the easiest type of data sharing to implement because the system does very little for the user. It requires a minimum amount of communications, usually only a file transfer protocol and sometimes data conversion. However, from the organization's perspective this approach can be very complicated because the scheduling and data integrity usually depends on manual, administrative procedures. The more frequently data must be exchanged and the more organizational units that are involved, the more error prone the procedure becomes.

This type of data interchange usually occurs when an organization is beginning to link two or more previously decentralized sites.

6.4.3 Shared Data.

The third approach involves shared data. This approach still has local control and use of the data, but, as with interchange data, other organizational units also need to use the data. The *difference is that with shared data conceptually or logically all of the organizational units are* using or sharing the same copy of the data, whereas with interchanged data each unit has its own independent copy of the data. Although the units sharing the data are using the same logical copy of the data, depending on the implementation of the system they may or may not be using the same physical copy of the data. The key point is that if different physical copies are involved, the system automatically and transparently maintains consistency among the copies. This means that except for performance, all of the units appear to be sharing a single centralized copy of the data.

There are variations of this data sharing approach depending on how the various units use the data. *One variation allows only the originating or control unit to enter and modify the data, while* all of the other units use it only for retrieval. For example, salesmen, purchasing, and other departments may be allowed to query the inventory data, but only a warehouse may be allowed to actually modify the inventory data. Another variation allows multiple units to directly modify the data, e.g., salesmen may be allowed to commit inventory for their orders and in doing so directly modify the inventory data. The critical factor in deciding how the data is to be shared involves business policy, not information systems technology. However, given the current state of the art, some policies will be easier or harder to support and may require more or less specialized applications.

The other issues involve coordinating the control of shared data. Conceptually, a single organizational unit must be responsible for the definition and control of the shared data. This includes defining the data both logically and physically, specifying the integrity constraints for the data, authorizing how the data will be shared, and scheduling, periodically updating, and maintaining the data. With local and interchange data this control presents no problem because one copy of data is the sole responsibility of one organizational unit. However, when two or more units share data, this control must be coordinated. In cases where one unit is clearly the dominant user, for example the only one authorized to update the data, the control may be very similar to local control with the exception being that it accommodates the needs of secondary users. When data are extensively shared by many units, for example when many units can update the data, then the coordination becomes much more complex.

Ideally, agreements can be negotiated among the various users. Where such agreements cannot be worked out among the sharing units, there are two options. One option is that if the data cannot be shared, then they must be interchanged. While this option is technically easy, it is not very desirable because it places an administrative burden on all of the organizational units that need the data. The other option is that if negotiations fail, then another unit makes the control decisions *by arbitration. This arbitrating unit is normally higher in the hierarchy.*

The previous discussion of spheres of control has assumed that all of the organizational units are peers. This is clearly not the case, since organizational units always exist in a hierarchy. When there is a disagreement at lower levels, the hierarchy is frequently invoked to resolve them. In some cases all of the lower level units report to the same higher unit. For example, if the regional marketing units cannot decide how to share the necessary data, then corporate marketing may arbitrate a solution and specify how the sharing will be done. Conceptually, what this does is convert shared data into local data by changing the organizational unit responsible for it. In other cases the units that *need to share the data do not report to the same higher level unit.* In these cases the data remain shared, but the higher units, which have a different organizational perspective, may be able to negotiate an agreement whereas the lower level units with their own more limited perspectives could not.

There is clear difference in the number of organizational units and spheres of control depending on whether an organization is centralized or decentralized and whether its information systems organization is centralized, decentralized, or distributed. This will become more apparent in the next two sections describing the starting points from which an organization begins to move toward distributed systems and the *planning that is necessary for transition.*

There is a clear trend toward more spheres of control and for more shared data. From the organizational perspective, the trend toward more local autonomy and control is creating more local units and allowing them more control over their data requirements. From the information systems side the trend is driven by cheap computers and LANs which allow more units to get their own processors and storage. As more data are originated and controlled by these local units with their own systems, the need to exchange and/or share data will become more important.

6.5 STARTING POINTS

Although there are many different types of distributed processing and levels of distributed database management, a distributed system is a goal or direction in which many organizations are moving for the various reasons identified earlier. However, an organization's migration path and how quickly it can move toward a distributed environment are determined by its starting point. This section describes the two starting points from which an organization can begin this migration.

6.5.1 Centralized.

One frequent starting point is a centralized system. This involves a single central site for hardware and data storage, for operations, and usually for the systems analysis, design, implementation, and maintenance. The only remote facilities and communications involve terminals for transaction processing and remote devices for job entry and output.

There are major differences, however, in how information resources are managed in a centralized environment. These differences also have a major affect on how an organization can progress. At one extreme, there are organizations that still take the traditional applications-oriented approach to data. They still do not see data as a critical resource to be managed. Correspondingly, their perception of a distributed environment includes very little information resource management.

At the other extreme there are organizations with a very sophisticated understanding of IRM. For these organizations a distributed environment will include policies and procedures, whether centralized or distributed, to maintain control and coordination of their information resources.

Between these two extremes there are organizations that have begun to understand the importance of information resources and manage them. For example, they may have a database management system and a database administration function, only in its more technical sense. These organizations will understand and be concerned with many of the technical issues involved in controlling and administering their data in a distributed environment, but they may not yet be sensitive to some of the more complex organizational issues, which become even more difficult in a distributed environment.

In summary, when an organization prepares to migrate from a centralized to a distributed environment, the most critical factor is its level of sophistication and understanding of information resource management. If it does not have effective control of its data resources in a centralized environment, it will have much more difficulty trying to establish the necessary control, either as part of the migration process or after it has shifted to a distributed environment.

6.5.2 Decentralized.

A decentralized environment involves multiple computer sites with virtually no communications between them. This situation frequently evolved in large organizations when geographically dispersed divisions computerized various operations independently, often with incompatible hardware and software. Depending on how a company is organized, these divisions may be doing different or similar functions. For example, divisions may be performing the same functions but for different product lines.

The easiest way to consider the decentralized starting point is to consider each site as essentially an occurrence of the centralized model. As with the centralized starting point, a key issue is the way in which a site manages its information resources. A key difference with the decentralized model is that different sites can, and probably will, be at different levels of sophistication in how they manage their information resources. This means that the different sites will have different levels of data quality and different expectations in terms of ease of use, development tools, and support facilities and in terms of the amount of control and coordination that is enforced. These differences will affect how rapidly the organization can migrate to a distributed environment and what that environment will be.

6.6 TRANSITION PLANNING

This section describes some of the issues involved in planning an organization's transition to distributed database management and IRM. There are both technical and organizational/administrative issues. The technical issues are essentially the same regardless of the organization's starting point, but the approaches and alternatives for the organizational and administrative issues may differ greatly. This section considers both types of issues, but emphasizes the organizational and administrative ones.

The technical issues involve most of the traditional database administration functions. These functions include:

- Selection and acquisition of a DBMS.
- Designing and defining the database.
- Controlling access to the database.
- Ease of use tools for improving database availability.
- Backup and recovery procedures.

Selection and acquisition may be easier with the centralized starting point because there is more flexibility in making system decisions. With the decentralized cases, systems are already in place, and the distributed system must frequently include them. In these cases, communications interfaces and data standards and conversion procedures must be formalized, both in terms of requirements and as availability tools.

Physical design of the database is much more complicated. The logical design of the database should not be affected by the question of distribution and there are several automated tools to support this effort. However, the physical design is much more complicated and today there are few tools to ease this effort. The users at the various sites must specify what data they need, how they want to access, and their requirements for response time and availability. Given these design requirements, how the data is actually distributed should be transparent to the users. This may actually be the case when the transition is from a centralized to a distributed system. However, if the starting point is a decentralized system, then the current way the data are located at the various sites may constrain how they are located in the distributed system.

Access also becomes more complicated. Users will need to be granted access to remote sites if they need data that are only available at those sites. If there are multiple copies of the data, are the access controls the same for all of the copies or are users only authorized to access certain copies for load balancing? Many of these issues will be easier moving from a centralized starting point because a whole new set of rules and expectations can be formulated. However, certain procedures and expectations are already in place in a decentralized system and it may be more difficult to modify them.

Ease-of-use tools such as high level query languages and data dictionary/directory systems will be even more important in a distributed environment because there will be less face-to-face interaction among the users and it may be harder to find a person who can answer questions about the way data are structured and accessed. Also, in a heterogeneous system, transparency tools to simplify the user interfaces and hide the differences between the various systems will be important.

Finally, backup and recovery in a distributed system will be more complicated. Failure at one site should not cause a failure or the loss of data integrity at other nodes. Although full automated recovery is the ideal target, coordinated administrative actions at several sites may sometimes be needed. This may mean that sites will have to coordinate their manual administrative backup and recovery efforts.

The basic organizational and administrative issues are the same regardless of whether the starting point is the centralized or the decentralized one. These issues involve information planning and standards. The same type of planning and standards are needed in either case, but the way they are developed and enforced will probably depend on the starting point.

If the centralized organization has an effective IRM function, then the transition may be relatively smooth. The necessary planning and standards functions will be in place in the centralized environment. The centralized IRM organization can decide which functions should be distributed, plan the necessary training, and work with the new IRM organizations at the distributed sites. In effect, the transition is then controlled and paced by the central IRM organization to ensure that adequate control is always maintained. Obviously, this transition will not be smooth if there is not an existing and effective IRM function at the central site. In this case the organization has two options. First, it can initially develop the central IRM function and then distribute it. Second, it could develop a plan for implementing the IRM functions and then simultaneously implement them at all of the sites. However, the first alternative probably has the greater chance of success. Any attempt to first distribute the data and then try to implement IRM is probably doomed to failure.

Organizationally, the transition from a decentralized environment is more difficult. First, there are many organizations (the IRM organization at each site) trying to maintain their control, and possibly having different perceptions of distributed IRM. Second, there may be different levels of understanding and sophistication about IRM at the various sites. Developing a common, accepted IRM transition plan in these cases will be very difficult. However, it can be done, given enough time and the necessary organizational skills.

6.7 DISTRIBUTED DATABASE MANAGEMENT TECHNOLOGY

This section describes some of the key technology issues for distributed database management systems (DDBMSs). It identifies several major dimensions along which to classify DDBMSs. Using these dimensions, it explains the options an organization has, given the current state of the art. It then describes the trends along which DDBMSs are developing, and reviews the functions the CODASYL Systems Committee has proposed for a DDBMS.

6.7.1 The State of the Art.

There are six dimensions along which a DDBMS can be classified. These dimensions involve:

- Data distribution.
- Location transparency.
- Update synchronization.
- Backup and recovery.
- Degree of homogeneity.
- Site autonomy.

The first dimension involves the type of data distribution the system supports. A DDBMS may allow only partitioned, non-redundant data, completely replicated data, or a hybrid database, distributed with any degree of replication. With partitioned data, the global database is divided into non-overlapping fragments and each fragment is placed at only one node in the network. With complete replication each node has a complete copy of the entire data. The most complex approach involves the hybrid distribution option, which allows any number of copies of each fragment. These same distribution alternatives also apply to the metadata. These distribution alternatives affect each of the other dimensions.

The second dimension is location transparency, i.e., does the user need to know where the data is stored. Only the earliest systems forced the user to specify where the data was located. Transparency is desirable for both ease of use and for data independence. Depending on how the data is distributed, it is easier or harder for the DDBMS to support this transparency. With fully replicated data there is no problem—all of the data is at every node. With hybrid data distribution and certain types of partitioning the DDBMS needs to consider both the data items requested and their actual values to determine where the data is stored. For example, just requesting inventory data does not give the DDBMS enough information to locate the data if each warehouse maintains its own inventory database. The system also needs to know which warehouse to query, or, as a default, it must query them all. This requires a more complicated data directory, when compared to simpler partitioning schemes where the data location is determined solely by the data item name, e.g., inventory level.

The two most difficult technical issues in distributed database management involve update synchronization and backup and recovery. Data partitioning, which allows only one copy of the data, dramatically simplifies these problems, especially if updates are not allowed to span nodes. Depending on an organization's precise requirements, simplifications are sometimes possible to make these problems manageable. For example, if all of the sites do not need absolutely current data, then the dominant copy approach allows relatively efficient updates of either replicated or hybrid data. With this approach, users can retrieve data from any copy (unless they explicitly request the most current or dominant copy), but all updates are routed to and controlled by a specific node, i.e., the one with the dominant copy. This minimizes the update synchronization overhead because locking the dominant copy implicitly locks all copies. However, the dominant copy can become a bottleneck if extensive updating is required. Another benefit is that this approach allows users to get the latest version if they need to. With some synchronization algorithms the concept of the latest copy does not exist. Today, there is no adequate solution to the update synchronization problem for the general case.

The degree of homogeneity is an important dimension. This involves both hardware and software or DBMS homogeneity. Most of the initial research prototypes of DDBMSs took the homogeneous approach, which simplifies the problem by eliminating the data and command translation. More recent work, however, has focused on heterogeneous DDBMSs. Heterogeneity is particularly important for decentralized organizations that are approaching distributed database management by connecting existing systems.

A final dimension involves site autonomy (in the technical, not the organizational sense), i.e., the degree to which the DDBMS affects the local DBMS and its operations. In a homogeneous DDBMS this is not an issue because all of the sites have the same DBMS and use the same algorithms. However, in a heterogeneous environment site autonomy is important to avoid the modification of existing DBMSs. Therefore, it is desirable to isolate in a separate layer the new functions, which are required because the database is distributed.

Given these dimensions, relatively limited capabilities are available today. The simplest method for dealing with distributed data is file transfer. Clearly, it is the most basic underlying mechanism on which more sophisticated capabilities can be built, but it does not really address the above dimensions.

A second, more sophisticated approach involves extracting data from a database and using a file transfer mechanism to move the data to another node. This is of particular interest in the micro-computer and workstation environment. A database can be queried, with the results being routed to another node for further processing. In fact, with some systems the data can be extracted along with its definition, downloaded to another node, and then reloaded into a local database for further processing. However, this approach has two major limitations. First, it involves an interchange copy of the data, not shared data. Changes to the original data or the downloaded copy are not reflected in the other copy. Second, this approach is normally used only for retrieval. To maintain database integrity, many organizations do not allow this mechanism to be used to update the central database. Although it is not an inherent limitation of this approach, today these extracts are only done against a single site, centralized database, not against a distributed database.

The third approach is for the organization to define its requirements and build its own DDBMS. Although this has been done in the past for special applications, it is not a viable option except in very limited cases, and even then only for very sophisticated organizations.

The last approach is to use some of the recently introduced, limited purpose distributed DBMSs that several vendors have announced. Most of these systems have specific limitations, but they are clearly steps in the right directions. Most of these DDBMSs are limited by the type of data distribution they support and the way in which they support updating.

6.7.2 Trends.

The basic trend is toward a more complete distributed database management system on which to build distributed applications. The assumption is that a full DDBMS must include both local DBMS functions (essentially all of those provided by today's centralized DBMSs) and an additional set of functions that are needed because of the distributed environment. The complete DDBMS functions can be packaged as an integrated set of software or the new functions may be packaged as an additional layer to be added on top of existing DBMSs, which would continue to operate as the local DBMS. The CODASYL Systems Committee report A Framework for Distributed Database Systems: Distribution Alternatives and Generic Architectures has identified five additional functions that are needed for such as system:

First, it must provide the linkage between the user and the local DBMS. Second, it must be able to locate the data in the network. This means that, given the logical data request, the DDBMS must be able to use the network data directory to determine where the data are stored in the network. Third, it must select the strategy to use in processing the request. This involves identifying alternate strategies and evaluating them. This is one of the major areas of future DDBMS development. Over time, DDBMSs will become able to accept more complex requests and develop strategies for processing them. The fourth function involves network-wide backup and recovery. The fifth function arises in a heterogeneous environment. This is a translation function to allow the DDBMS to convert both data and requests between different systems.

A final development which must occur, although there is little indication of it yet, involves design aids and development tools for distributed systems and distributed databases. Without such tools these systems will continue to be labor intensive, one of a kind creations.

6.8 SUMMARY

This chapter has reviewed the basic concepts of IRM and identified several key extensions, including managing unstructured data (such as text, images, and voice), capturing more data semantics, managing external data, and coordinating the communications resources in a distributed environment. It identified the types of objects (i.e., technology, data, policy, and function) that can be distributed and described some of the distribution alternatives. It then identified the various factors encouraging and inhibiting the trend toward distributed processing.

Using the concept of "spheres of control," it described several ways in which data can be controlled and shared in a distributed environment. These methods involved local data, interchange data, and shared data, with various levels of distributed database management support.

The chapter then described the centralized and decentralized structures from which most organizations begin their migration toward a distributed system and the necessary transition planning. Finally, it reviewed the current state of the art and the trends which will encourage even more organizations to move in the distributed direction.

IEEE COMPUTER SOCIETY

1730 Massachusetts Ave., N.W.
Washington, D.C. 20036-1903
U.S.A.

Non-profit
Organization
U.S. Postage
Paid
Silver Spring, MD
Permit No. 1398