NIST Special Publication 500-167

Computer
Systems
Technology

U.S. DEPARTMENT OF
COMMERCE
National Institute of
Standards and
Technology

Information Management Directions: The Integration Challenge

Elizabeth N. Fong
Alan H. Goldfine

The National Institute of Standards and Technology[1] was established by an act of Congress on March 3, 1901. The Institute's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Institute conducts research to assure international competitiveness and leadership of U.S. industry, science and technology. NIST work involves development and transfer of measurements, standards and related science and technology, in support of continually improving U.S. productivity, product quality and reliability, innovation and underlying science and engineering. The Institute's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the National Computer Systems Laboratory, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; provides calibration services; and manages the National Standard Reference Data System. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Computing and Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[3]

The National Computer Systems Laboratory

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Laboratory consists of the following divisions:

- Information Systems Engineering
- Systems and Software Technology
- Computer Security
- Systems and Network Architecture
- Advanced Systems

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-cutting scientific themes such as nondestructive evaluation and phase diagram development; oversees Institute-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following divisions:

- Ceramics
- Fracture and Deformation[3]
- Polymers
- Metallurgy
- Reactor Radiation

[1] Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.

[2] Some divisions within the center are located at Boulder, CO 80303.

[3] Located at Boulder, CO, with some elements at Gaithersburg, MD.

NIST Special Publication 500-167

Information Management Directions: The Information Challenge

Elizabeth N. Fong and Alan H. Goldfine, Editors

Information Systems Engineering Division
National Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

September 1989

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

NIST

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) (formerly the National Bureau of Standards) has a unique responsibility for computer systems technology within the Federal government. NIST's National Computer Systems Laboratory (NCSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. NCSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. NCSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports NCSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

Library of Congress Catalog Card Number: 89-600755
National Institute of Standards and Technology Special Publication 500-167
Natl. Inst. Stand. Technol. Spec. Publ. 500-167, 174 pages (Sept. 1989)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1989

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

PREFACE

This report constitutes the proceedings of a 3 day workshop on Information Management Directions held in Fort Lauderdale, Florida on October 31 - November 2, 1988. The workshop was the fifth in the Information Management Directions series (formerly called Data Base Directions) sponsored by the National Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), formerly the National Bureau of Standards, in cooperation with the Association for Computing Machinery (ACM), the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE), and the Federal Data Management Users Group.

The first workshop in this series was Data Base Directions: The Next Steps, held in October, 1975. It addressed the question: "What information about database technology does a manager need to make prudent decisions about using this new technology?"

The second workshop, Data Base Directions: The Conversion Problem, was held in November, 1977. It addressed the questions: "What information can help a manager assess the impact a conversion will have on a database system?" and "what aid will a database system be during a conversion?"

The third workshop, Data Base Directions: Information Resource Management—Strategies and Tools, was held in October, 1980. It considered information management tools from the standpoints of: uses; policies and controls; logical and physical database design.

The fourth workshop, Data Base Directions: Information Resource Management—Making It Work, was held in October, 1985. It assessed the nature of information resource management practice and problems, and reported on solutions that have proven workable.

The fifth workshop, called Information Management Directions: The Integration Challenge, is the subject of this report. This workshop focused on issues related to integration and productivity.

The workshop divided into five working groups to consider: (1) the integration of knowledge and data management, (2) the integration of technical and business data management, (3) the integration of systems planning, development, and maintenance tools and methods, (4) the integration of heterogeneous computing environments, and (5) architectures and standards. Each group prepared a draft report that was then put into final form by the proceedings editors.

Because the participants in the workshop drew on their personal experiences, they sometimes cited specific vendors and commercial products. The inclusion or omission of a particular company or product does not imply either endorsement or criticism by NIST.

We gratefully acknowledge the assistance of all those who made the workshop a success.

Elizabeth Fong, Editor
Alan Goldfine, Editor

TABLE OF CONTENTS

Page

EXECUTIVE SUMMARY

On October 31 - November 2, 1988, the National Computer Systems Laboratory of the National Institute of Standards and Technology, formerly the National Bureau of Standards, in cooperation with the Association for Computing Machinery Special Interest Group on Management of Data, the IEEE Computer Society Technical Committee on Database Engineering, and the Federal Data Management Users Group, held the fifth in the series of Information Management Directions (formerly called Data Base Directions) workshops. The purpose of these workshops is to examine in depth key trends and strategies that affect the future of the information management profession. The focus of Workshop 5 was on issues related to integration and productivity.

The need for **Integration** has become a challenge for the information management profession. The dictionary definition of "integration" is "the condition of being formed into a whole by the addition or combination of parts or elements." The aspect of integration that was covered in this workshop related to the formation of the "whole" information management discipline through the addition or combination of the various related technologies.

The workshop was organized into five working groups, which met to discuss:

- the integration of knowledge and data management

- the integration of technical and business data management

- the integration of systems planning, development, and maintenance tools and methods

- the integration of distributed, heterogeneous computing environments

- architectures and standards.

The keynote speaker, Tom DeMarco of the Atlantic Systems Guide, spoke on "Standardization: An Oblique View." DeMarco claimed that standards do more harm than good when they work against the prevailing culture, and that the essence of standardization is discovery, not innovation.

The Integration of Knowledge and Data Management

During the last several years, the information industry has experienced a gradual convergence of artificial intelligence and database management. This panel, chaired by Robert Curtice of Arthur D. Little, was charged with discussing the factors underlying this convergence, and examining the implications of the convergence for the next generation of application systems and databases. The perspective was to be that of both the users and the builders of these future applications.

The panel identified some approaches to integrate artificial intelligence technology and database management technology to achieve knowledge base management. The advantages and disadvantages of each approach were analyzed.

The panel examined the implications of the convergence, and concluded that there are impediments to the integration of knowledge and database management. The global integration impediments identified were: existing investments in applications, different application development cultures, different application development methodologies and tools, lack of trained personnel, and the need for definition of new jobs, roles, and organizations.

The Integration of Technical and Business Data Management

The manufacturing industry has traditionally separated technical data and business data. However, during the last half decade there has been increasing pressure to integrate technical and business data into a common data management environment. This panel, chaired by T.N. Bernstein of the U.S. Air Force, was charged with examining the key issues related to building a common environment for the management of technical and business data.

The panel first considered the integration problem in terms of differences in the nature of data, differences in the nature of systems, differences in maturity of methodologies, and organizational and cultural issues.

The panel developed a step-by-step process to achieve integration. The process starts with establishing goals for an integration effort and formulating an integration plan. In developing an integration strategy, standardization is one of the principal tools and, where possible, standards, guidelines, and conventions should be used. The process requires iteration and incremental refinement, and finally a "public relations" program to inform, educate, and solicit the cooperation of all affected members of the organization.

In answering the question, "what benefits can the user expect to gain from the integration of technical and business data management?" the panel proposed a set of metrics to assess the real impact of an integration project on the corporation.

The panel used a framework to point out both the end results of the integration process and the system considerations for achieving those results. Finally, the panel provided a prognosis of what is likely to emerge in the near future that will affect the possible end results of integration.

The Integration of Systems Planning, Development, and Maintenance Tools and Methods

Hundreds of methodologies and automated tools have been developed to support the many different tasks involved in systems planning, development, and maintenance. For the most part, these methods and tools were developed independently, but true productivity in information management can only come from an effective integration of specific methods and tools in specific environments. This panel, chaired by John Zachman of IBM, was charged with discussing the basic issues surrounding the selection and integration of tools and methods to achieve overall productivity.

The panel used a framework to serve as a basis for discussion. This framework consists of a matrix whose columns represent Data, Process, and Network, and whose rows represent Objectives, Business Models, Information Systems Models, Technology Models, Detailed Representations, and Functioning Systems. The eighteen cells of this matrix provided the general focus for analyzing information systems architecture. Integration was then analyzed according to three perspectives: horizontal integration, across specific rows of the matrix; vertical integration, across specific columns; and intra-cellular integration, across the entire scope of the enterprise from the perspective of a single cell.

The panel concluded that current information systems are dis-integrating, not integrating. Only through integration is it possible to achieve advances in productivity, asset leverage, quality, flexibility for assimilating changes, and so on.

The Integration of Heterogeneous Computing Environments

The current computing environment in most businesses can be defined as being distributed and heterogeneous. During the last decade, telecommunications technology has evolved to make it possible to link these heterogeneous machines and their resident applications and databases to one another. However, the overall objective is not just to establish telecommunications links among various machines and databases to enable them to talk to one another. It is to establish an environment that is transparent to users, and that provides them with real-time access to data wherever and however it is stored. This requires the development of a new kind of integration technology, centered around data management. This panel, chaired by Sandra Heiler of Xerox Advanced Information Technology, was charged with discussing the problems of information integration, focusing primarily on semantic issues, and with reviewing the mechanisms for producing federated systems.

The panel explored various aspects of heterogeneity in computing environments and considered various models of integration that range from loose coupling of autonomous components to tight coupling or "integrated" systems.

The panel then presented an architecture for federated systems that comprise heterogeneous components, focusing first on the meta-model and schema, then on integration services that must be provided outside the local systems.

The panel concluded that integration is based on agreements between participants in a federation. The group felt that physical integration is a nearly solved problem because formal methods of specifying agreements at this level exist. However, logical integration is much harder because there are no formalisms for expressing semantics. More research is needed in meta-models, but perhaps some candidates are the object/function models.

The panel also concluded that the appropriate mode of coupling systems is probably not at the extremes of loose or tight coupling, but some place in the middle. Integration services for loosely coupled systems consist of providing generic facilities while allowing components to remain autonomous. Integration services for tightly coupled systems must be based on knowledge of the internals of specific components.

Architectures and Standards

This panel, chaired by W. Bradford Rigdon of McDonnell Douglas Information Systems, was charged with addressing the role of architectures and standards in supporting information management throughout an enterprise.

The panel first identified five levels of architecture: business unit, information, information system, data, and delivery system. The panel then discussed the problems, benefits and risks to an enterprise of an architecture, and developed some specific examples of standards within different levels of the architecture.

The panel concluded that standards should be used to implement and enforce the architecture. There is not a single correct way to develop an architecture or implement standards for every enterprise; they must be customized to the environment.

INFORMATION MANAGEMENT DIRECTIONS: THE INTEGRATION CHALLENGE

Elizabeth N. Fong
Alan H. Goldfine
Editors

ABSTRACT

This report constitutes the results of a 3 day workshop on the integration challenge of information management, held in Fort Lauderdale, Florida on October 31 to November 2, 1988. The workshop was sponsored by the National Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), formerly the National Bureau of Standards (NBS), in cooperation with the Association for Computing Machinery (ACM), the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE), and the Federal Data Management Users Group.

This workshop was the fifth in the Information Management Directions series (formerly called Data Base Directions). The purpose of these workshops is to examine, in depth, key trends and strategies that affect the future of the information management profession. The focus of this fifth workshop was on issues related to integration and productivity.

The 72 workshop participants were organized into five working panels, which met to discuss the integration of knowledge and data management; the integration of technical and business data management; the integration of systems planning, development, and maintenance tools and methods; the integration of distributed, heterogeneous computing environments; and architectures and standards for information management.

Key words: architecture; database; database management; DBMS; distributed; information management; integration; knowledge base; standards.

1. INTRODUCTION: THE PARADOX OF INTEGRATION

Daniel Appleton

GENERAL CHAIRMAN

Biographical Sketch

Daniel S. Appleton is Chairman and CEO of D. Appleton Company, Inc. (DACOM). He specializes in industrial modernization and data resource management.

Since founding DACOM in 1979, Mr. Appleton has continued to play a leadership role in defining the sociological and technological issues related to managing information as a true enterprise asset. His conceptualization and systemization of Information Asset Management (IAM)—which focuses on data as the quintessential information asset—has received world-wide acclaim from the highest levels of management to line operatives, as well as in academic circles. It has also been broadly accepted throughout the manufacturing industry where modernization of information management is crucial to business survival.

As a result of his practical thinking, Mr. Appleton has been personally influential in formulating many IRM and CIM programs for Fortune 500 businesses and for structuring several significant programs, including:

- o The Department of Defense Computer-Aided Acquisition and Logistics Support (CALS) initiative
- o The U.S. Air Force Integrated Computer-Aided Manufacturing (ICAM) Program
- o The Computer and Automated Systems Association (CASA) of the Society of Manufacturing Engineers

Mr. Appleton has published numerous technical papers and articles, and he is the most published author in Datamation magazine, having had more than 20 articles published. He is a world recognized lecturer on IRM and manufacturing modernization.

"Every complex system," say Briggs and Peat in their book **Turbulent Mirror**, "is a changing part of a greater whole, a nesting of larger and larger wholes leading eventually to the most complex dynamical system of all, the system that ultimately encompasses whatever we mean by order and chaos—the universe itself."

This observation, despite its metaphysical overtones, describes the foundation of the dilemma we know as integration. There is an assumption, based on the intuitive acceptance of complex systems as constituent parts of an integrated universe, that integration is good. It keeps the universe together. Not only is it good, it is inevitable. It is the natural end game of a universe that is constantly struggling to keep itself whole.

This notion, of course, flies in the face of quantum physics which, since its beginnings in 1900, has made a habit of thrashing Newtonian mechanics. Newtonian mechanics assumes that the universe is simply a vast integrated algorithm, all of the functions and variables of which are calmly working their way with us. At the foundation of quantum physics is the second law of thermodynamics, which states that entropy increase is irreversible. To put it another way, quantum physics rests on the assumption that the universe is not struggling to maintain its natural condition as an integrated whole; it is committed to a course of self-destruction; it is dis-integrating.

Is integration a natural state of the affairs of the universe, or is it an unnatural condition? This is the essence of the integration paradox. The resolution to this paradox lies in the reality that it takes energy to maintain or to establish the condition of integration. Thus, we have a situation where we are constantly forced to exert energy to sustain or create integration. If we are to do this, there must be some reason for exerting this energy, some reason whose benefits outweigh the costs of the energy itself. In other words, since integration has a cost, it must have a greater benefit, or else it will not occur.

What is the benefit of integration? Well, historically we have seen integration as a condition of orderliness. Things that are dis-integrated are also dis-orderly and chaotic. Chaos in business is expensive. Integration, therefore, is an economically motivated business objective. And it is expensive, because it must occur in a universe which, because of the second law of thermodynamics, is hostile to integration.

"Information technology is a misnomer." "Information technologies" are what we have. In fact, we have a blizzard of technologies, with no effective means of ordering them. These technologies are wholes unto themselves. They are not parts of a whole. They are, in fact, dis-integrated, and the growing business perception is that this condition is becoming chaotic. Every corporation on this earth is struggling to create a new order out of the chaos of dis-integrated information technologies. And, this new order is expensive. It is not the natural state of affairs. The integration of information technologies into a coherent information management strategy for business is something that must be justified on its benefits; i.e., the reduction of chaos. It is natural, therefore, that this workshop be dedicated to the issue of the integration of information technologies.

But, why do we need a general workshop on information technology integration? Aren't the vendors of these technologies seeking an integrated solution? Can't we just wait until the integrated solution is ready for us to buy? NO! Technology vendors feed on a condition of dis-orderliness because the alternative—order—preconditions the process of selection. This preconditioning is an economic promise to us, and an economic threat to many vendors.

The raison d'etre of the National Institute of Standards and Technology (NIST) is to facilitate the maintenance of order through the sponsorship of standards activities. NIST also has in its charter the challenge of stimulating the development of new technologies. It is a fine balance that must be maintained between the development of new technologies and the integration of those technologies into economic investment alternatives for businesses. This fine balance is nowhere more evident than in the area of information technology.

The purpose of the Information Management Directions workshop was to approach the issue of integrating information technologies into a comprehensive structure. The outcome of the workshop was the determination that integration, itself, is a technology. But, it is not just a technology in the sense of bits and bytes; it is also a management technology. Further, this workshop concluded that the natural foundation for the integration of information technologies into a comprehensive information management strategy for business revolves around a single common denominator: data. Data, it turns out, is the common structure that underlies the bits and bytes technology and the management technology of the integrated information environment. It is the foundation of integration technology.

The Information Management Directions workshop was comprised of working groups that examined both aspects of integration: 1) bits and bytes, and 2) management. The bits and bytes issues were studied by: the panel on the Integration of Heterogeneous Computing Environments; the Integration of Technical and Business Data Management panel; and the panel on the Integration of Knowledge and Data Management. The management side of integration technology was examined by the Integration of Systems Planning, Development, and Maintenance Tools and Methods panel and the Architectures and Standards panel.

While these panels operated somewhat autonomously, they were, in fact, integrated. The integrated conclusions are very revealing. They are as follows:

1. While the technologies under study are disparate in their origins and objectives, they contain many elements in common. Technical data management and business data management, for example, while seemingly different technologies, share solution logics that are greatly similar. The same is true of expert systems and data management technology.

2. The solution logics in all of these technologies come together at the level of data management. Data, it seems, is a common denominator of all information technology.

3. The shift from technological solutions based on disparate problem definitions to solutions based on common technological elements represents a major paradigm shift in the world of information management. The new paradigm is, in a sense, the "industrialization of information technology."

4. The common data management technology that can be exploited to integrate technical, business, and knowledge base information technologies appears to be

what began as "relational data management" and is evolving into what has become known as "object oriented data management." These distinct data management technologies provide a common denominator for business computing, technical computing, and expert systems, and they also provide the common structure necessary for dealing with distributed, heterogeneous computing.

5. The focus on data management as the driver of information technology integration and the new information technology paradigm that it represents cannot be implemented without a major shift in the methodologies, and consequently the CASE tools, used by businesses for application planning, development, and maintenance. This shift requires a focus more on the "data driven" approach.

6. Architectures define an integrated end state vision for information technology within a specific business context. Standards are the harbingers of integration and at the same time they are an essential element of integration. Standards are the bridge among architectures, information technology, and applications management.

Like the universe, information technology began with a Big Bang. Like the universe, information technology has been expanding at the speed of light. Like the universe, information technology is locked in a life and death struggle with the inexorable force of entropy increase. And, like the universe, information technology can only survive if we expend energy to maintain and create order and structure. This means integration.

2. KEYNOTE ADDRESS

STANDARDIZATION (AN OBLIQUE VIEW) (Abstract)

Tom DeMarco

Biographical Sketch

Tom DeMarco is a principal and founder of the Atlantic Systems Guild of New York. He was the winner of the 1986 Warnier Prize for lifetime contribution to the field of computing and software.

His most recent work is PEOPLEWARE: Productive Projects and Teams (with co-author Tim Lister) published by Dorset House. He is also the author of Structured Analysis and System Specification (Prentice Hall, 1982), and of more than ninety articles and papers about management and the system development process.

Mr. DeMarco's career began at Bell Telephone Laboratories where he served as part of the now-legendary ESS-1 project. In later years, he managed real-time projects for La CEGOS Informatique in France, and was responsible for distributed on-line banking systems installed in Sweden, Holland, France, and Finland. He has lectured and consulted throughout the Americas, Europe, Australia, and the Far East.

Mr. DeMarco is the current president of MODUS, the Modula-2 Users Society. He has a BSEE degree from Cornell University, an MS from Columbia University and a diplome from the University of Paris at the Sorbonne.

1. "A new technology is half of a good thing ..." the other half is an accepting sociology. Technical innovation frequently fails or fails to achieve its promise because the accepting sociology is not there.

2. Standards do more harm than good when they work against the prevailing culture.

3. Excessive documentation is the most evident failure of standards, "Documentation is more often part of the problem than part of the solution." Our obsession with textual documentation adds costs, early binding, and excessive human drudgery to the life cycle.

4. Documentary standards tend to be cumulative: A new method imposed adds to the documentation burden; old documentation standards are carried along, unexamined. The impact of CASE: we add a picture to the documentation, but forget to remove the thousand words.

5. Any standard that adds to the documentation burden is likely to be counter-productive. Call for "Zero-Based Documentation." Two goals to strive for: 1) a document that won't be updated shouldn't be written in the first place; and 2) documentation should be a free or nearly free by-product of the process itself (we need to learn to live with the 'working documents' that the projects generates naturally).

6. The standardizers' dilemma: It's more amusing to innovate than to look for, and call attention to, de-facto standards. But the essence of standardization is discovery, not innovation.

3. THE INTEGRATION OF KNOWLEDGE AND DATA MANAGEMENT

Robert Curtice

CHAIRMAN

Biographical Sketch

Robert Curtice has been addressing problems in information processing, systems architecture, and database management for clients since he joined Arthur D. Little, Inc. in 1966. His consulting assignments have centered on the design methods, software, and other technical and managerial issues arising in the planning and development of database oriented computer systems. Mr. Curtice has worked with manufacturing, financial, transportation, utility, retail, and service organizations in the development of database plans and applications.

Mr. Curtice was the General Chairman of the fourth workshop in this series, Data Base Directions: Information Resource Management—Making It Work, in 1985.

He is the author of Standards and Procedures for the Data Base Administration Functions, Strategic Value Analysis: A Modern Approach to Systems and Data Planning, and is the co-author of Logical Data Base Design, Second Edition.

PARTICIPANTS

Susan Blair
Don Libes
Richard J. Mayer
Prasad Nair
Marlene A. Palmer

Tim Rau
Robert S. Rosenzweig
Stanley Y. W. Su
Alvin Underbrink

3.1 INTRODUCTION

Data management technology has been a part of the mainstream of information systems activities for many years. Generalized database management systems (DBMS) have been in commercial use for two decades. Research and development efforts in the data management arena continue to produce new concepts and products, (e.g., relational databases, distributed databases, semantic databases, object oriented databases).

Research and development in artificial intelligence (AI) has progressed in a parallel, yet distinct, fashion from data management technology. One definition of AI is that it is concerned with automating work previously done by people, by emulating human problem-solving behavior [FEI63]. Examples include duplication of natural skills (e.g., speech and pattern recognition), improved information systems development (use of AI technology), and duplication of learned skills and expertise (e.g., expert systems).

Recently, workers in the two fields of data management and artificial intelligence have discovered common ground and begun to identify a new area of endeavor: knowledge base management.

A knowledge base management system (KBMS) is a general-purpose software system that provides not only the database management functions provided by traditional database management systems such as data modeling facilities, query languages, transaction management, integrity and security control, concurrency control, recovery, etc., but also artificial intelligence systems functions such as knowledge representation techniques, deductive problem solving capabilities, enhanced user interfaces, explanation facilities, reasoning capabilities, etc. It is a new technology yet to be developed and is the result of integrating database and AI technologies. Many advanced applications such as computer aided software engineering (CASE), office automation, and computer integrated manufacturing (CIM) can benefit from this new technology.

3.2 BENEFITS OF KNOWLEDGE BASE MANAGEMENT

Benefits and uses of knowledge base management are understood within the broader context of information systems. All useful information systems involve encoded human knowledge that takes the form of:

- data
- decision rules
- operations

in the commonly understood meaning of these terms. The distinction between knowledge base and more traditional information systems is then one of degree rather than absolute. For example, knowledge base systems are more likely to

- contain generalized inference processors in order to exhibit complex human behavior
- be capable of recounting the inferences by which a result was arrived at
- be able to learn (self adaptive)
- involve heuristics, iterative search, and trial-and-error techniques
- be based on an explicit model of human behavior
- be able to determine inconsistencies not previously thought of.

The knowledge base itself consists of

- stored data
- operations
- inference rules

and the knowledge base management systems provide facilities for specifying and manipulating these objects. Both object-oriented database management systems and AI shells exhibit some of the characteristics of complete knowledge base management systems. Figure 3-1 provides a comparison of some of the key facilities of a KBMS, and indicates which of them are typically provided by traditional DBMS software, object-oriented systems, and AI shells.

Software System	DBMS	Object Oriented DBMS	AI Shell	KBMS
Facility to define and manipulate	DB2 Oracle	Gemstone V-Base	ART Kee	Loops? Statice-Joshua OSAM
Stored Data	✓	✓		✓
Integrity/Security Constraints	✓	✓		✓
User Defined Operations and Data Types		✓	?	✓
Application Constraints	?	?	✓	✓
Inference Rules and Engines			✓	✓
Other AI Features			✓	✓

Figure 3-1

Use of the complete facilities of a knowledge base management system will have the following benefits over more traditional approaches:

- o increased application functionality
- o cost savings to the business function supported
- o reduced application development cost
- o shortened implementation times
- o improved application maintainability.

3.3 INTEGRATION OF AI AND DBMS TO ACHIEVE KNOWLEDGE BASE MANAGEMENT

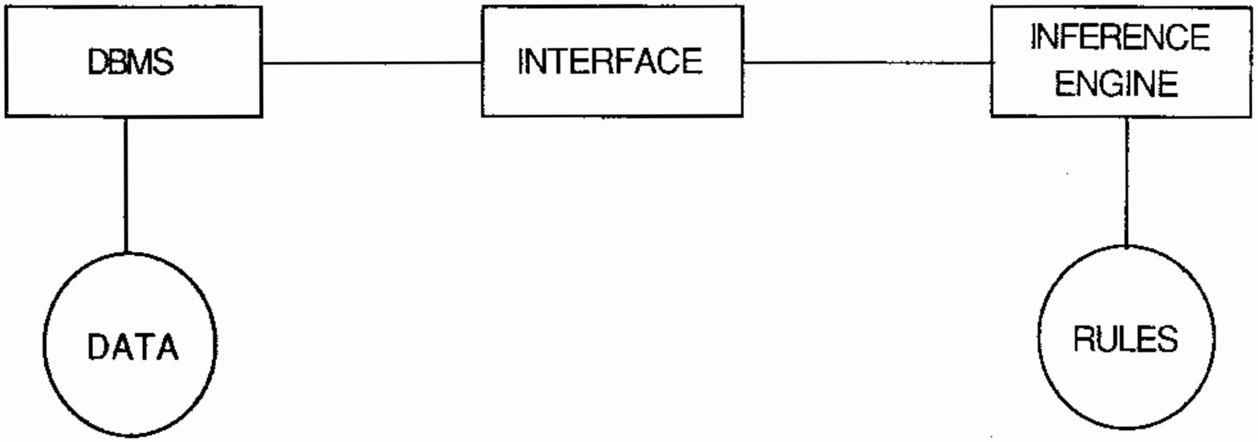
Several approaches for merging AI and DBMS technologies have been proposed [GAL83], [BRO84], [JAR84b], [VAS85] and [WHA87]. They are the bridging approach, the extension approach and the integration approach.

3.3.1 Bridging

This approach, illustrated in figure 3-2, establishes an interface (or bridge) between an existing DBMS and an existing inference system so that the functions provided by these two independently implemented systems can be combined and used to establish a KBMS. The DBMS is used to manage a database of facts or assertions (i.e., the extensional database) and the inference system is used to manage and manipulate deductive rules (i.e., the intensional database) in problem solving. Good examples of this approach can be found in [KEL82], [JAR84a] and [KEL84].

The main advantage of this approach is that it makes use of what is already available. The only software that needs to be developed is the interface between two existing systems. This approach is also particularly useful in many real-world environments in which many data and rule bases have already been established in the existing systems. It provides a simple and convenient way for accessing the available "knowledge" that has been captured by these systems.

Integration Approach -- Bridging



Advantages:
- o Independent testing and development
- o Systems and data/rules already available

Disadvantages:
- o Limited to power of DBMS or inference engine
- o Different data representation in two systems
- o Inference overhead between two systems

Figure 3-2

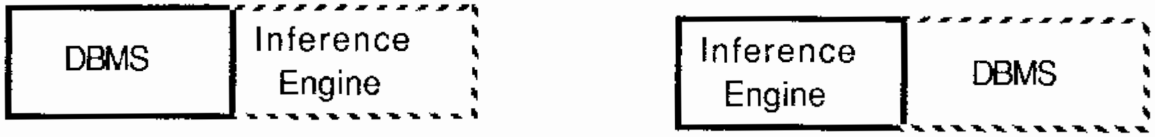
This approach also has a number of disadvantages. One disadvantage lies in the fact that there are two different knowledge representations, one for facts and one for rules. Conversion or translation of semantic contents between two systems is necessary and system efficiency can become a serious issue. The second disadvantage relates also to efficiency considerations. Since an inference plan in an inference process is created using the intensional database and is verified using the extensional database, ping-ponging between two systems by frequent program calls will take place. This may generate a considerable amount of overhead. The third disadvantage stems from the fact that two systems are functionally independent. As a result, the expressive power of rule specification in the inference system is limited by the rule specification language used (e.g., predicates or horn clauses). It is not possible, for example, to use the full power of a high-level query language to define the condition or consequent part of a deductive rule.

Lastly, functional separation between two systems may require that some of the functions be duplicated in both systems. For example, it was pointed out in [KEL84] that aggregation operations (e.g., Sum, Count, Average, Maximum and Minimum) that are commonly available in a DBMS and are useful in rule specifications, cannot be used to operate on deduced facts in an inference engine and thus have to be duplicated in both systems.

3.3.2 Extension

This approach, illustrated in figure 3-3, implements a KBMS by extending either a DBMS or an AI system to include the functionalities of the other. For example, one can enhance a DBMS with deductive and triggering power as demonstrated in [ST083], [WON84] and [STO84] or enhance a logic programming system with database facilities [WAR84]. The advantage of this approach is similar to the bridging approach. It takes advantage of an existing system to build the desired KBMS. Although this approach may achieve some of the necessary functionalities of a KBMS, it is difficult to extend an existing system to handle requirements that were not considered in the original system design and implementation. The resulting system may not be as well-structured, reliable and efficient as it could have been if the KBMS had been built from scratch.

Integration Approach -- Extension



Advantages: o Build on existing systems

Disadvantages: o Difficult to modify existing systems

o Integration may require fundamental change to system architecture

Figure 3-3

3.3.3 Integration

This approach, illustrated in figure 3-4, aims to integrate the required DBMS and AI functions in a new KBMS system architecture in which redundant functions can be avoided and data and rules can be better organized and more efficiently accessed. In this approach, integration can take place both at the representation level and the functional level. Integration at the representation level means that the KBMS provides a uniform representation framework for defining facts and rules.

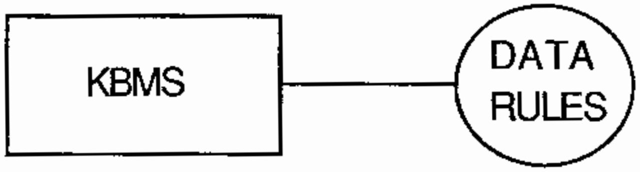
For example, the object-oriented paradigm is used in [WIE83], [KER84], and [WIE84], to define the structures, operations, and knowledge rules of the object classes of an integrated knowledge base. In [SU85], [RAS87] and [RAS88], a knowledge manipulation language is used for specifying both the users' queries and knowledge rules. At both compilation time and run time, rules can be activated to modify a user's query and be included as part of a transaction. Thus, the traditional techniques of a query processing, query optimization, and transaction management can be used for rule processing as well.

Furthermore, the full expressive power of a knowledge manipulation language can be used to express complex knowledge rules. Also, knowledge rules are naturally distributed among the object classes in a class lattice so that rules relevant to the objects in these classes are readily available when the objects are being processed. They are considered as an integral part of the object class definition.

Integration at the functional level means that operations for processing data and rules are functionally integrated so that functional redundancy can be avoided. Also, operations on data and rules can be intermixed. This is important since both forward and backward reasonings require intimate interaction between facts and rules.

The main disadvantage of the integration approach is that it cannot make use of any existing software. A new KBMS has to be developed from scratch.

Integration Approach -- Integrated



Advantages:

- o Unified approach to define process, data, rules
- o Efficient access to rules by distributing rules to object types
- o Data is readily available when processing rules

Disadvantages:

- o Few available implementations
- o Existing data and rules must be reloaded

Figure 3-4

3.4 GLOBAL INTEGRATION IMPEDIMENTS

Five major areas have been identified as impediments to the integration of knowledge and database management:

- existing investments in applications
- different application development cultures
- different application development methodologies and tools
- lack of trained personnel
- need for definition of new jobs, roles, and organizations.

3.4.1 Existing Investments in Applications

The investment in applications using current database management systems is enormous. Even moving an existing application from one DBMS to another cannot be undertaken lightly. Converting a major application running under IMS to a relational database management system is only undertaken after a penetrating analysis of the need and the economics of the opportunities; the effort is not trivial.

3.4.2 Different Application Development Cultures

AI solutions have usually attacked specific tasks from a "stand alone" point of view; information system professionals have come to address more and more "total system" solutions where components of the system work in concert and data is shared and not redundantly introduced. The system developers' increasing use of prototyping techniques, and the integration of them as components of large systems, should help prepare the information systems professional to think in terms of integrated knowledge base management systems. The users of the current AI and expert system technologies are increasingly participating in the development of information systems.

3.4.3 Different Application Development Methodologies and Tools

Current information resource dictionary systems (IRDSs) do not directly support the development of knowledge base applications. Their primary concern is to isolate the data and data structures required by the applications. The operations and inference rules and their manipulation, which have historically been imbedded into the application, can now be externalized through the use of specialized software such as generalized expert systems. There are as yet no design guidelines that help the application developer decide where information should be stored, i.e., when should information be stored as inference rules, data, operations, or code. The current family of CASE tools is not geared toward the concepts of KBMS. Until practical examples of CASE tools addressing the issues emerge, there is not likely to be an effort to design CASE tools for large application development efforts under an integrated knowledge base management system.

3.4.4 Lack of Trained Personnel

There are currently very few professionals who have strong information systems backgrounds and have been involved in knowledge base applications. This will begin to change. The situation is similar to the emergence of office systems and "stand alone" decision support systems. The availability of work stations, the realization that there are common data sources, and the acknowledgment that networks of people solve problems and carry out activities, drives toward integrated system solutions (loosely coupled). In consequence, the necessary skill base gets developed in solutions where accesses to extant databases are required and the cost of redundant entry of information prevents extending the use of the AI technologies.

3.4.5 Need for Definition of New Jobs, Rules, and Organizations

Five areas have been identified requiring modified or new definitions to support integrated knowledge base management systems: data administration, database administration, knowledge base administration, knowledge engineering, and the user. In today's information systems environment, the role of data administration and database administration is understood in terms of defining and managing the data needed to model the objects of concern (business or other) and performing functions using this data. In particular, database administration is concerned with the organization and sharing of the data. Depending on the sophistication of the organization, the rules for the operations using the data may be included in the data administration function. The scope and authority of the positions depend on the centrifugal and centripetal administrative forces throughout the organization.

The users, or non-system professionals, would appear to have a dominant role in determining where and how the administration of data, operations, and inference rules should take place; the systems professional would help to define the system issues that must be addressed by knowledge base administration. The clarification of the relationship of knowledge engineering to systems analysis, and of both to the user's role, needs to occur. In all these areas a lack of understanding and clear definition of functions and responsibility within the organization not only inhibits the effective use of integrated knowledge and database management systems, it also exacerbates the struggle over turf.

3.5 REFERENCES FOR CHAPTER 3

[BRO84] Brodie, M. L., and Jarke, M., "On Integrating Logic Programming and Databases," Proceedings of the First International Workshop on Expert Databases, October 1984.

[FEI63] Feigenbaum, E. A., and Feldman, J., Computers and Thought, McGraw-Hill, New York, 1963.

[GAL83] Gallaire, H., "Logic and Databases," Panel discussion, Proceedings of the International Joint Conference on Artificial Intelligence, 1983.

[JAR84a] Jarke, M., Clifford, J., and Vassiliou, Y., "An Optimizing Prolog Front-end to a Relational Query System," Proceedings of the ACM SIGMOD Conference, June 1984.

[JAR84b] Jarke, M., and Vassiliou, Y., "Coupling Expert Systems and Database Management Systems," in Artificial Intelligence Applications for Business, Ablex, Norwood, NJ, 1984.

[KEL82] Kellogg, C., "Knowledge Management: A Practical Amalgam of Knowledge and Database Technology," Proceedings of the National Conference on Artificial Intelligence, Carnegie-Mellon University, August 16-20, Pittsburgh, PA, 1982.

[KEL84] Kellogg, C., Kogan, D., Travis, L., Van Buer, D. and Whitney, R., "Knowledge Management," submitted to SIGART Special Issue on AI and Data Base Research, 1984.

[KER84] Kerschberg, L., and Shepherd, A., "Constraint Management in Expert Database Systems," Proceedings of the First International Workshop on Expert Database Systems, October 1984.

[RAS87] Raschid, L., The Design and Implementation Techniques for an Integrated Knowledge Base Management System, Ph.D. Dissertation, Department of Electrical Engineering,, University of Florida, Gainesville, FL, 1987.

[RAS88] Raschid, L., and Su, S.Y.W., "A Transaction Oriented Mechanism to Control Processing in a Knowledge Base Management System," Proceedings of International Conference on Expert Database Systems, April 1988.

[STO83] Stonebraker, M., Woodfill, J., and Andersen, E., "Implementation of Rules in Relational Database Systems," Database Engineering, Volume 6, Number 4, December 1983.

[STO84] Stonebraker, M., Kung, R., Hanson, E., Ioannidis, Y., Sellis, T., and Shapiro, L., "Heuristic Search in Database Systems," Proceedings of the First International Workshop on Expert Database Systems, October 1984.

[SU85] Su, S.Y.W., and Raschid, L., "Incorporating Knowledge Rules in a Semantic Data Model: An Approach to Integrated Knowledge Management," Proceedings of the AI Application Conference, Miami, FL, December 1985.

[VAS85] Vassiliou, Y., "Integrating Database Management and Expert Systems," Proceedings of the Conference on

Data Base Systems for Office, Engineering and Science, Karlsruhe, West Germany, March 1985.

[WAR84] Warren, D.S., and Sciore, E., "Towards an Integrated Database-Prolog System," Proceedings of the First International Workshop on Expert Database Systems, October 1984.

[WHA87] Whang, K.Y., and Navathe, S.B., "An Extended Dysjunctive Normal Form Approach for Processing Recursive Logic Queries in Loosely Coupled Environments," IBM Technical Report, IBM Watson Research Center, Yorktown Heights, New York, 1987.

[WIE83] Wiederhold, G., Milton, J., and Sagalowicz, D., "Applications of Artificial Intelligence in the Knowledge-Based Management Systems Project," Database Engineering, Volume 6, Number 4, December 1983.

[WIE84] Weiderhold, G., and Missikoff, M., "Towards a Unified Approach to Expert and Database Systems," Proceedings of the First International Workshop on Expert Database Systems, October 1984.

[WON84] Wong, E., Ioannidis, Y., and Shinkle, L., "Enhancing INGRES with Deductive Power," Proceedings of the First International Workshop on Expert Database Systems, October 1984.

3.6 SUGGESTIONS FOR FURTHER READING

Integration of Artificial Intelligence and Database Technologies

Al-Zobaidie, A., and Grimson, J.B., "Expert Systems and Database Systems: How Can they Serve Each Other?" Expert Systems, Vol. 4, No. 1 (February 1987), pp. 30-37.

Bell, D.A., "An Architecture for Integrating Data, Knowledge, and Information Bases," Informatics 8, Advances in Intelligent Retrieval: Proceedings of a Conference Jointly Sponsored by Aslib, the Aslib Informatics Group, and the Information Retrieval Specialist Group of the British Computer Society, Oxford, England, April 1985, pp. 240-257, London, England, Aslib, 1985.

Gardarin, G., and Gelenbe, E., eds., New Applications of Data Bases, London, England, Academic Press, 1984.

Guida, G., and Tasso, C., "An Expert Intermediary System for Interactive Document Retrieval," Automatica, Vol. 13, No. 6 (1983), pp. 759-766.

Hancock, P.A., and Chignell, M.H., eds., Intelligent Interfaces: Theory, Research, and Design, Amsterdam, North Holland, 1989.

Harris, L.R., "The Marriage of AI and Database Technology," AI Expert, Vol. 2, No. 3 (1989), pp. 7-8.

Hushon, J.M., and Conry, T.J., "The Evolution of Gateway Technology," Proceedings of the 2nd Conference on Computer Interfaces and Intermediaries for Information Retrieval, Boston, MA, May 1986, Office of Information Systems and Technology, Defense Technical Information Center, Alexandria, VA, 1986, pp. 181-201.

Jakobson, G., "Toward Expert Database Front-Ends," Proceedings of the 2nd Conference on Computer Interfaces and Intermediaries for Information Retrieval, Boston, MA, May 1986, Office of Information Systems and Technology, Defense Technical Information Center, Alexandria, VA, 1986, pp. 317-318.

Jones, K.P., ed., Informatics 7: Intelligent Information Retrieval: Proceedings of a Conference Held by the Aslib Informatics Group and the Information Retrieval Group of the British Computer Society, Cambridge, England, March 1984, Aslib, London, England, 1984.

Keller, R., Integrating Artificial Intelligence: Managing Expert System Technology, Yourdon Press, New York, 1986.

Kowalik, J.S., ed., Coupling Symbolic and Numerical Computing in Expert Systems, Elsevier Science, New York, 1986.

Monarch, I. and Carbonell, J., "CoalSort: A Knowledge-Based Interface," IEEE Expert, (Spring 1987), pp. 39-53.

Mylopoulos, J., and Brodie, M., Readings in Artificial Intelligence and Databases, Morgan Kaufmann Publishers, Palo Alto, CA, 1989.

Pederson, K., "Connecting Expert Systems and Conventional Environments," AI Expert, Vol. 3, No. 5 (May 1988), pp. 26-35.

Risch, T., Repon, R., Hart, P., and Duda, R.A., "Functional Approach to Integrating Database and Expert Systems," Communications of the ACM, Vol. 31 (1988), pp. 1424-1437.

Sparck-Jones, K., and Boguraev, B.K., "A Natural Language Front End to Data Bases with Evaluative Feedback," New Applications of Data Bases, Academic Press, London, England, 1984, pp. 159-182.

Worden, R., "Integrating KBS into Information Systems--The Challenge Ahead," Knowledge-Based Systems: Applications in Administrative Government, Ellis Horwood Ltd. on behalf of the Central Computer and Telecommunications Agency, Chichester, UK, 1988, pp. 23-33.

Knowledge Base Systems

Beeri, C., ed., Third International Conference on Data and Knowledge Bases, Morgan Kaufmann Publishers, Palo Alto, CA, 1988.

Brodie, M., and Mylopoulos, J., eds., On Knowledge Base Management Systems, Springer-Verlag, New York, 1986.

Cleal, D.M., and Heaton, N.O., Knowledge-Based Systems: Implications for Human-Computer Interfaces, Ellis Horwood Ltd, Chichester, UK, 1988.

Davis, R., "Knowledge-Based Systems," Science, Vol. 231, No. 4741 (February 12, 1986), pp. 957-963.

Davis, R., and Lenat, D.B., Knowledge-Based Systems in Artificial Intelligence: Two Case Studies, McGraw-Hill, New York, 1982.

Davis, R., Shrobe, H.E., and Szolovits, P., Readings in Knowledge-Based Systems, Morgan Kaufmann Publishers, Palo Alto, CA, 1989.

Duffin, P.H., Knowledge-Based Systems: Applications in Administrative Government, Ellis Horwood, Ltd., on behalf of the Central Computer and Telecommunications Agency, Chichester, UK, 1988.

Edosomwan, J.A., "Ten Design Rules for Knowledge Based Expert Systems," Industrial Engineering, Vol. 19, No. 9 (August 1987), pp. 78-80.

Ercegovac, Z., "Knowledge-Based Expert Systems: A Profile and Implications," Proceedings of the 5th National Online Meeting, New York, NY, April 1984, Medford, NJ, Learned Information, Inc., 1984, pp. 39-46.

Garner, B.J., "Expert Systems: From Database to Knowledge Base," Information and Software Technology, Vol. 29, No. 2 (March 1987), pp. 60-65.

Ginsberg, A., Automatic Refinement of Expert System Knowledge Bases, Morgan Kaufmann Publishers, Palo Alto, CA, 1988.

Kellog, C., "From Data Management to Knowledge Management," IEEE Computer, Vol. 19, No. 1 (January 1986), pp. 75-84.

Mockler, R.J., Knowledge-Based Systems for Management Decisions, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Mockler, R.J., Knowledge-Based Systems for Strategic Planning, Prentice Hall, Englewood Cliffs, NJ, 1989.

Ow, P.S., and Smith, S.F., "Two Design Principles for Knowledge-Based Systems," Decision Science, Vol. 18 (Summer 1987), pp. 430-447.

Sriram, D., Knowledge Based Expert Systems, Computational Mechanics, Billerica, MA, 1987.

Sriram, D., and Adey, R.A., eds., Knowledge Based Expert Systems for Engineering: Classification, Education, and Control, Computational Mechanics, Billerica, MA, 1987.

Ullman, J.P., Principles of Database and Knowledge-Base Systems, Computer Science Press, Rockville, MD, 1988.

Voelker, J.A., and Ratica, G.B., "The Genesis of a Knowledge-Based Expert System," IBM Systems Journal, Vol. 25, No. 2 (1986), pp. 181-189.

Walker, A., "Knowledge Systems: Principles and Practice." IBM Journal of Research and Development, Vol. 30, No. 1 (January 1986), pp. 2-14.

Walker, A., et al, eds., Knowledge Systems and Prolog: A Logical Approach to Expert Systems and Natural Language Processing, Addison-Wesley, Reading, MA, 1987.

Winstanley, G., Program Design for Knowledge Based Systems, John Wiley & Sons, New York, 1987.

Object-Oriented Computing

Cox, B.J., Object Oriented Programming: An Evolutionary Approach, Addison-Wesley, Reading, MA, 1986.

Meyer, B., Object Oriented Software Construction, Prentice-Hall, Englewood Cliffs, NJ, 1988.

"Object-Oriented Languages Reorient Programming Techniques," Computer Design, Vol. 26, No. 20 (November 1, 1987), pp. 52-62.

Parsaye, K., Chignell, M., Khoshafian, S., and Wong. H., Intelligent Databases: Object Oriented and Deductive Hypermedia Technologies, John Wiley & Sons, New York, 1989.

Shriver, B., and Wenger, P., Research Directions in Object-Oriented Programming, MIT Press, Cambridge, MA, 1987.

Stefic, M., and Bobrow, D., "Object-Oriented Programming: Themes and Variations," AI Magazine, Vol. 6, No. 4 (1986), pp. 40-62.

Stein, Jacob, "Object-Oriented Programming and Databases," Dr. Dobb's Journal, March 1988, pp. 18-34.

Yonezawa, A., and Tokoro, M., eds., Object-Oriented Concurrent Programming, MIT Press, Cambridge, MA, 1987.

Zdonik, S.B., and Maier, D., eds., Readings in Object-Oriented Databases, Morgan Kaufmann Publishers, Palo Alto, CA, 1988.

4. THE INTEGRATION OF TECHNICAL AND BUSINESS DATA MANAGEMENT

T. N. Bernstein

CHAIRMAN

Biographical Sketch

Mr. Bernstein is the program manager for the Air Force's Integrated Design Support System, an advanced development program to create an information system concept for acquiring and disseminating critical engineering data for Air Force weapon systems. Mr. Bernstein received his B.S. degree in Aeronautical Engineering from the University of Cincinnati. He has planned, directed, and performed research in structural analysis and design, primarily computer aided design, and has spent the last 10 years in information resource management.

PARTICIPANTS

Olin Bray
Jean-Pierre DeJean
Mary Ann Engelbert
Lawrence E. Frank
Robert E. Fulton
Larry L. Johnson

Randall Johnson
Mary Loomis
Shamkant B. Navathe
Stephen Ritzman
John F. Willis

4.1 INTRODUCTION

In recent years there has been a dramatic increase in the amount of data used within organizations that are concerned with design, development, modification and operation of complex technical products. These products include aerospace vehicles, transportation systems, energy facilities, computers, and buildings. Traditionally, data has been separated into technical data and business data. Separate organizational and system capabilities have been developed to support the management of these two sets of data.

It has become increasingly apparent that the separation of these two islands of technical and business information is not in the best interest of the organizational enterprise. Accordingly, steps toward integration of technical and business data are needed. This report summarizes the issues and findings of a study on the ramifications of integrating business and technical data.

To clarify the issues it is useful to define **information integration** as used herein. Information integration is the establishment of the appropriate computer hardware/software, methodology, and organizational environment to provide a unified and shared information management capability for a complex business enterprise. A measure of the effectiveness of information integration is whether a user can get the right data, at the right place, at the right time, in the right form, and at the right cost. The user's limitations are based only on his need to know, store, modify, and otherwise access the data. Key elements of information integration include an infrastructure composed of:

- an information architecture with
  - single logical database
  - location transparency
  - physical transparency
- an organization to control and manage the information and take advantage of the available integration tools
- a software architecture and set of appropriate tools
- a hardware architecture of appropriate levels of computers and peripherals.

The benefits of information integration are many, and include such areas as:

- o cost reductions through
  - -- increased productivity
  - -- reduced time for locating data
  - -- minimized data redundancy
- o reduced product development time
- o improved quality through
  - -- reduced development changes
  - -- improved change control
- o streamlined operation
- o formalized communication.

This study investigated the characteristics and end results of information integration, the process and metrics of integration, and the techniques and tools of integration. The report contains a summary of key findings and details of the areas studied.

4.2 BASIC FINDINGS

The panel on Integration of Technical and Business Data concluded its discussion with the following four major findings:

4.2.1 Characteristics of the Data

The characteristics of technical data and business data differ in degree.

Data Characteristics:

The superficial view of business data being character-oriented and technical data being graphics oriented was promptly set aside. A table showing the various characteristics of technical and business data was constructed. Upon review of this table, the panel reached the conclusion that technical and business data differ from one another only in a matter of degree.

Data Management:

Technical data management requires mechanisms that are not required in most current business applications. These mechanisms include version and configuration control, support of rich data types, handling of high volumes of data, access to the data at different levels of abstraction to support a variety of presentation capabilities, support of rigorous release procedures, and authentication of the data. However, some office data (for example, "what-if" analysis) seem to be more closely aligned in its data management requirements with technical data than with business data.

Data Transactions:

Technical data and business data have markedly different transaction duration and rate requirements. Further, the consistency problem, as viewed by the applications, can be tackled through different approaches. For example, version control in the technical case versus locking in the business case.

Database Management:

The traditionally more demanding data management requirements of technical data (data types, multiple levels of abstraction, authentication, version control) will be of use in the advanced business applications of the future. This will further lessen the perceived differences between business and technical data. It is questionable as to whether a single database management technology could satisfactorily meet the functional and performance needs of both business and technical data systems. If so, then perhaps the distinctions between technical and business data will be seen as insignificant and immaterial.

4.2.2 Organizational Considerations

Organizational considerations are significant determinants of the success of an integration project.

Eliminate sub-optimal (parochial) solutions:

Typical integration efforts uncover duplication of efforts, and recommend shifts in workloads and responsibilities to resolve the problems resulting from suboptimization in the existing organizations. Each organization typically has been organized and automated without concern for the overall strategic (profit driven) goals of the corporation.

Involve users:

Early involvement of the users in the definition of the functions and data needs of the business (as-is and to-be) and in the future allocation of business functions in the to-be system is mandatory for the success and acceptance of an integration project. Integration must be viewed as a means to leverage existing business advantages of the corporation while lessening known shortcomings. Through integration, the organization is presented with the opportunity to reduce cost and cycle time, to improve quality, and to identify new products and market opportunities.

Users are jointly responsible for success:

The responsibilities for the success of any integration project rest squarely with the user organizations acting as the primary system through the integration of their business functions. The integration of the information systems, supporting the newly redefined business functions into a cohesive whole, then becomes an implementation step of secondary concern.

Roles of the Information Systems Organization:

The Information Systems organization should not be viewed as the primary beneficiary and sole proponent of integration. Instead, it should be viewed as a co-consultant, focused on solving a business rather than information system problem. The primary role of the Information Systems organization is:

- to facilitate the system analysis (as-is, to-be) of the user organization operations

- to ensure that the newly integrated system complies with the corporate information and computer architectures

- to define and evolve the information and computer architecture capable of supporting the overall business goals of the corporation.

Top management support:

Top management support was identified as a key contributor to the success of integration. This support is crucial in undertaking projects that transcend traditional organization boundaries, and that exceed, in scope and duration, typical Information Systems project norms. Top management support is also required to keep systems compliant with the corporate information and computer architectures as systems evolve in response to the changing business and technical environments.

Gaining top management support:

Support from top management can only be expected when top management is convinced that integration serves the strategic (primary concern of top management) rather than the tactical goals of any one organization. In particular, integration must not appear to be a goal only of the information systems portion of the organization. Top management must understand that integration yields competitive advantages that competition cannot duplicate. This is as opposed to the ease with which stand-alone products can be acquired and deployed. This occurs because integration leverages the existing competitive position (market information, technical skills, installed assets) of the organization.

Top management commitment is not simply measured by the resources (schedule, manpower) allocated to integration. It is mainly expressed by the willingness to make the organizational trade-offs promoting and maintaining inter-operable corporate systems. Top management must look at integration of the business functions as a significant corporate asset.

4.2.3 Project Life Cycles

The project life cycle is important in setting proper management directions.

A prime issue related to the successful definition and implementation of integration in a corporation is the widely shared consensus that integration activities are difficult to justify to management. Management is viewed as being accustomed to evaluating the merits of a project through traditional financial measurements. Unlike hard automation projects, for which management has intuition, integration demands the commitment of resources to activities (e.g., functional and data modeling) that have no appeal to management. Further, benefits of these activities are difficult to foresee at the beginning of the investment period.

Essential elements of a successful integration project are detailed below:

Define an open computing and information architecture:

Integration efforts must begin with the definition of a computing environment (hardware, network, system software, system standards), and an information definition methodology (functional, data modeling methodologies, and integration techniques). It must also have the means of controlling the transition of existing applications and systems into an integrated computing environment.

The architecture must be closely aligned with the major industry standardization efforts. This is necessary to allow for the orderly growth of the computing environment (new applications, new technologies, new standards), and to support unforeseen demands for inter-operability, both from within a corporation and from the marketplace (anticipation of more requirements like CALS in the defense and non-defense markets).

Identify and tackle visible problem areas that will benefit from integration:

Integration can be implemented in a bottom-up fashion in visible areas of the business where cost/benefits, quality improvements, cost, and cycle time reduction can be realized and demonstrated to management. This bottom-up approach can be utilized as long as a well-thought-out information and computer architecture is available and is used to provide technical directions (e.g., standards, methods, tools) during implementation activities.

Develop metrics to assess the benefits of integration:

Given the difficulty in quantifying benefits in advance, the anticipated benefits require that before integration, productivity metrics be acquired to evaluate the success of the integration project throughout its life cycle.

Likewise, it is important to establish reasonable expectations among the user organizations. These expectations must be documented in memoranda of understanding reflecting the commitments of the user organizations to the integration effort.

Involve users early in the project:

This is the key to a generation of truly useful ideas yielding significant business benefits. Justification of integration through savings in data processing costs is unlikely to be significant. This is because data processing costs amount to only a few percent of the design and manufacturing cost of a high technology product. The mechanisms recommended to obtain early user involvement are:

- tackling a pressing user business need through integration
- joint development of the as-is and to-be functional and data models of the user business.

4.2.4 Integration Metrics

The panel developed four sets of metrics to evaluate the presence, or absence, of an integrated information system. These metrics cover the following major areas:

- data integration
- organization integration
- system integration
- identified financial benefits resulting from integration.

The metrics are both qualitative and quantitative measurements of the success displayed by a corporation in defining, implementing, and maintaining an organization structure, a computer architecture, and an information architecture that bring about identified profits to the corporation through novel use of information management.

4.3 THE INTEGRATION PROBLEM

The integration problem is defined across the following categories:

- differences in the nature of business and technical data
- differences in business and technical systems
- differences of maturity in business and technical methodologies
- organizational and cultural issues.

4.3.1 Differences in the Nature of Business and Technical Data

To assess the problem of integrating business and technical data, we began by evaluating the difference in the basic nature of the two data domains. These potential differences are important because:

- Integration of the two domains will require data management facilities that can accommodate the data from both domains where necessary.
- Since virtually all commercial DBMSs were developed to support business applications and data, it follows that fundamental differences between technical and business data would render the current offering of DBMS tools useless for technical data, producing a serious impediment to integration of the two data domains.

In searching for fundamental differences, we assessed a list of fundamental data characteristics. The panel was looking for those that are not required by one domain or the other, or which were fundamentally different in some sense. Among the data characteristics examined were:

- standard data types
- extended data types
- relationships and relationship types

- schema characteristics
- data volume
- data sharing
- data integrity
- data distribution
- data semantics
- data versioning and configuration management
- paradigms of interoperability, interchangeability, and behavior.

Although the investigation was begun with the common presupposition that there are fundamental qualitative differences between technical and business data, we could find no such differences between the two domains. The fundamental differences found were quantitative in nature. A few of the important quantitative differences are discussed below.

- Standard data types. The "exotic" data types encountered in technical data are also arising in office automation concepts as office integration proceeds.
- Extended data types. The requirement for per-application data types, which originates in object-oriented programming concepts, is seen as defeating integration, which focuses on persistent objects coordinated under a global schema. That is, data types that cannot be recorded, coordinated, and communicated cannot be shared.
- Relationships and relationship types. Both business and technical data require a fundamental capability in expressing relationships and relationship types. Technical data, however, tends to have a richer set of relationship types, and extremely complex relationships among data objects.
- Schema characteristics. Technical data (particularly in a business such as electronics manufacturing) will have a richer vocabulary of objects and relationships than business data. Additionally, technical schemas

will tend to change more frequently, tracking new technology and standards; whereas business schemas are more stable. However, as mentioned, this is a matter of degree.

- Data volume. Both business and technical data involve large amounts of data. Technical data items tend to aggregate other data items into much larger objects than is customary for business data. In these early days of engineering information models, entire design "databases" are often tracked and managed as a single data item measuring from megabytes to terabytes in magnitude.

- Data versioning and configuration management. Versioning is only modestly used in business data, but is an extremely important feature in technical data. Versioning is not well supported in current DBMSs.

- Paradigms of interoperability and interchageability and behavior. These paradigms illicit a strong feeling that technical and business data are fundamentally different. This is due to their requirement for extremely frequent application in the technical domain, and its occasional use in the business domain. In business, a purchase order is predictably relatable to a relatively small number of entity-types. The technical community (electronics design in particular) requires applying the paradigms heavily. An electrical part is relatable to a large number of entity-types, i.e., any type of electrical part (there are a myriad). An electrical part, furthermore, also serves a role as a mechanical part that must be fit together with other physical (mechanical) entity-types. Once a design has been made (a relationship of electronic and mechanical objects), its behavior must be simulated. Therefore, the occurrence of a part-type in a design must also contain its behavior. This type of complexity is the forte of object-oriented approaches seen increasingly and frequently in CAD tools today.

  It is felt that the business domain will require increased use of these paradigms as office automation proceeds into such areas as hyper media, etc. However, it is unlikely that it will make the heavy use of them that the technical domain demands.

The observation then, is that business and technical data differ primarily in a quantitative rather that qualitative fashion at both the level of instantiation and at higher levels of abstraction, including metadata. This implies that technical data and business data will be able to be moved among technical and business systems. This does not imply, however, that the systems currently in use in business are, or will be, adequate for technical use or vice versa. The question as to whether a single data system architecture would ever prove suitable to both domains was left open. It was felt that the domains currently require different architectures, but that the data will be migrateable where necessary.

4.3.2 Differences in Business and Technical Systems

In addition to data characteristics, the following system characteristics were examined for differences:

- Operating environment. The operating environment of business systems tends to be focused on large mainframes that are tightly controlled.

  Technical systems are often found distributed on small workstations with little or no formal control. The operating systems on which they are hosted are diverse.

- Transaction size. Transactions on technical systems can, by orders of magnitude, consume more time and involve more data than business systems. Business systems often quantify performance in terms of transaction response time. Technical transactions can be as brief as business transactions, or take many hours to execute.

- Public (standard) models. Business oriented DBMSs enjoy standardization (e.g., CODASYL and SQL). Business data models tend to be uniform within a company, but very diverse among different companies.

  The DBMSs of technical data are often proprietary and incompatible from tool to tool. The DBMSs are individually designed to meet the demands of complexity and the demands of interactive graphic-based editing. However, there are a number of efforts underway to standardize the conceptual models against which some of these tools operate, e.g., EDIF, VHDL, PDES, and CALS.

- Application types. Applications in business systems are generally designed against an external data model. Multiple applications have little trouble in operating against the same data.

  Applications in technical systems are generally designed against a proprietary representation of data. Multiple tools from different vendors will not, in general, operate on the same data without a conversion of format. Consequently, data conversion utilities are frequently encountered when multi-vendor tools are needed to operate on a design.

- Data distribution. When they are distributed, business systems tend to be tightly coupled, behaving more or less as a single system.

  The data distribution in technical systems is broad and loosely coupled. There is, in general, no effective way to coordinate the distributed data beyond the islands of servers mentioned below.

- Data sharing. Generally, the entire body of business data is interpretable by a single system.

  Technical workstations consist of diverse tools running on diverse platforms, with few of the tools understanding the data of another. Data sharing requirements have led to islands of small-scale centralization implemented as servers. The servers, like the workstations, are diverse and seldom are able to take advantage of one another as networked servers.

Essentially, the problem of integration from a systems standpoint is one of accommodating data across a great diversity of platforms, and the migrateability of data across multiple formats, data organizations, and levels of control. It was the general feeling that homogenization of the technical and business domains is doomed to failure, but that each could adapt to and learn from the other. Any system integration approach will have to accommodate those features of each that have evolved out of necessity.

4.3.3 Differences of Maturity in Business and Technical Methodologies

The methodology used to conduct business in most companies has evolved over literally hundreds of years. The automation of the methodology was one of the first targets of early computing, making it mature and well defined.

Technical methodologies, such as mechanical or electrical design, are generally not formalized to an adequate degree, making the construction of external data models and sufficiently formalized procedures difficult.

It is felt that formalization of technical methodologies is a critical step toward integration of the various technical domains themselves, let alone the integration of technical and business domains.

4.3.4 Organizational and Cultural Issues

It was generally agreed that the integration of business and technical domain data is as much a problem of organization and culture, as it is a problem of technology.

The way technologists regard their data is different than that of their counterparts in the business domain. In the business world there is seldom any difficulty in viewing data as belonging to the company, with access granted according to a well established methodology of doing business. The data produced by technologists, however, is often a product of their personal creativity, causing a strong sense of personal ownership. Trusting this data to "the system" is often not an easy step for a technologists to make. When the data is submitted to "the system," it is not uncommon to find that copies remain behind, cluttering up disks with files and shelves with floppies. This redundancy seldom serves well as the intended safeguard, but more often frustrates configuration management efforts when the role of "master" gets confused between the submitted and the shelved data.

As mentioned above, standardization and documentation of the technologists' methodologies is essential in an integration program. Technologists often resist attempts at standardization for fear of having their creativity overencumbered by "rules and regulations." There is a danger in this. However, it can be avoided. The capture of the methodology must include levels of control and coordination to accommodate "the creative spirit" early in the design cycle. However, as it becomes necessary to share work among technologists, whereby one's work is dependent on another's, the data must be brought under control, or the team members will find themselves working at odds with one another. The level of control must be escalated as the product definition moves toward production. Convincing the technologists that the standardization will not needlessly encumber them will be at least as difficult as defining the methodology itself.

Existing organizations have evolved around existing ways of doing things. The way things are done in an integrated environment is different from the way they are done in an environment that is not integrated. Imposing integration over an existing organizational structure can result in "mixed signals." For example, it may be necessary to incur greater costs in one functional area or department in order to achieve great offsetting savings across many other departments. If budgetary goals and evaluations do not reflect the necessity of the increased costs, then the integration program will be undermined by the conflicting goals of the departments.

Parochialism is the enemy of an integration program. Such parochialism will exist whenever it is possible for individuals to advance by making their departments look good at the expense of overall detriment to the corporate whole. Criteria must be carefully established so that an individual or department looks good only when the corporate goals are being met.

One often finds a pervasive lack of understanding between technologists and their business counterparts. Bringing the two groups to a mutual understanding and appreciation of one another's particular problems and point of view is important to the integration effort, particularly if the business systems group is to address the data integration needs of the technologists. In many companies, two data processing support groups have arisen due to the differences in the business and technology cultures. Properly managed, such an arrangement can aid an integration effort, but is probably not a necessary feature of an integration program.

4.4 PROCESS TO ACHIEVE INTEGRATION

4.4.1 Establishing Goals for an Integration Effort

Integration is not something any company or manager wants for its own sake. What they desire are the benefits derived from integration.

To obtain a commitment from management, the benefits to be derived must be enumerated in the terms of the business. The problems of the current operational environment that can be addressed by integration should be identified with their possible savings. Further, they should be ranked according to the amount of the savings. If possible, costs associated with the solutions should be approximated. However, the principal objective is to identify and rank the problem areas according to potential payback from integration. Unless the organization has had some experience in integration, the identified potential benefits will tend to be conservative and the costs elusive. Case studies of other companies that have suffered similar problems and successfully solved them will be useful.

Goals must be established for each of the problem areas. Each goal should have reasonable metrics associated with it. It is through these metrics that progress towards the goal is to be measured. The goals must be stated in the terms of the business itself, rather than in terms of data systems technology.

The goals must be prioritized according to the amount of potential payback, so that the initial focus is on the departments or functions of the organization that suffer most from the lack of integration. In general, the members of these organizations will be the most enthusiastic and cooperative players in the integration effort.

Some example goals are:

- reduce cycle time from product specification to production
- reduce scrap caused by use of the wrong revision of numerical control programs
- eliminate incorrect or untimely parts delivery to the assembly line.

All of these goals have metrics associated with them that often are already being tracked and used by the corporation.

In the case of broad, general goals, it will be useful to establish subgoals (milestones with corresponding metrics), and an understanding of the relationship of each subgoal to its broader goal. Often a subgoal will contribute to the accomplishment of a number of broad goals. This may indicate a high priority subgoal.

4.4.2 Formulating an Integration Plan

The formulation of the integration plan involves a formal analysis of the organization and the integration goals. The plan is generally devised by systems engineers and operations analysts working closely with management and technologists. This team must start with, or acquire, an intimate knowledge of the workings of the corporation.

The plan focuses on "what" has to be done to achieve the goals, rather than how things are to be accomplished. The "hows" are addressed primarily in the integration strategy. The only analysis of the "hows" done in this planning stage is to support cost estimation.

Typical deliverables of the analysis of goals are:

o Information Model. This is the identification of data and materials that are used in common by the organizational units to be integrated. During the planning stage, the information model will only be as detailed as is required to determine data bottlenecks and islands that are inhibiting the ability of the company to function efficiently.

o Analysis of Flow. The organization and functional units of the business must be modeled with attendant flow of information among the units.

o Methodology Documentation. This is the set of standard policies and procedures that govern the production of the organization's products and/or services. The methodology needs to be examined for information flow requirements. When it comes time to do this, many are surprised at how much of their business is governed by an "oral tradition" that has escaped the scrutiny of operational analysis.

- o Organization Study. This is the determination of how a company's organization helps or hinders the flow of information required to accomplish its principal objectives (the production of its products, services, etc.)

Having completed the analysis of the "as-is" situation; models should be constructed to reflect candidate "to-be" scenarios.

The integration plan must address the "granularity" of integration in terms of technical and organizational issues. This involves identifying what data is to be controlled in the integrated environment and when in the data's evolution it is brought under control. For example, controlling design data too early can place a needless burden on the system and may impede its development; controlling it too late will defeat many of the benefits integration has to offer.

The plan must also address how far down in the organization the integration is necessary. Often it will be found that functional areas have small, well-defined data overlap with other areas. Each area can implement its own sub-plan of information integration and control as long as it meets the requirements of an "umbrella" plan that governs the data sharing among them.

The "to-be" models must then be analyzed to assure that they will meet the established goals of the effort. Cost/benefit analysis must be performed on the "to-be" models. In the early stages of the integration effort these estimates will be rough, but will be refined over time.

The objective is to rank the goals of the effort to revisit the goals of the effort; adding, deleting, and refining. The goals and the implementation plan are iteratively refined until acceptable "as-is" and "to-be" models have been produced.

4.4.3 Developing an Integration Strategy

A strategy must be developed through which the integration plan is to be accomplished.

One of the foremost issues in forming the strategy is determining what needs to be standardized to achieve the plan's objectives. Standardization is one of the principal tools of an integration strategy. Where possible, public standards, guidelines, and conventions should be used.

It is generally necessary to standardize information models, interchange protocols, data representations, etc., but that is not the total extent to which standardization should be considered. It is useful to standardize platforms (hardware, operating systems, and communication software). However, it has been noted that the challenge of integration is typified by a great diversity among these items. The root of this diversity is basically a diversity in requirements for different data applications. It is often found, however, that the diversity of the systems in use is broader than dictated by the requirements of the business. It is best to "standardize" on a minimal set of platforms selected to address the requirements of various aspects of the business.

Cultural and organizational issues must also be addressed in the strategy. Although a sound technical strategy is necessary for a successful integration program. It is not sufficient by itself. Too often the organizational and cultural aspects of an integration strategy are underemphasized or ignored entirely, severely hobbling or even dooming the effort. An elegant technical solution will not succeed if its use cannot be accepted and assimilated by the culture of the company. User readiness for each step in the strategy must be assessed before it is implemented. When necessary, interim technological solutions that are short of what is possible can be inserted to allow a more gradual evolution of the system.

The strategy should include a project management strategy through which short-, mid-, and long-term goals can be tracked. The tasks that need to be performed to accomplish the goals are identified and detailed with all dependencies noted. The tasks are then costed in detail and critical issues and risk are assessed.

The results of strategy formulation may well impact the integration goals and integration plan from which it was derived. The goals and the plan should be revisited if this is the case.

4.4.4 Iteration and Incremental Refinement

The end of each of the sections on plan and strategy state that the previous "step" may have been impacted and will have to be revisited. This deserves emphasis. Integration involves a great deal of refinement. During the execution of the integration tasks, much will be learned about the problem as it specifically relates to the organization. This new knowledge must be folded into the plan repeatedly.

It is not feasible to envision a total solution from the outset. An organization must start with what is known at the time, and refine the plan iteratively. Although one must be on guard against constant changes in direction, there will be a number of false starts in different areas that will shed light on the problem at hand, further defining it. Management needs to know that a certain amount of this is expected as necessary and desirable.

Frequent reviews of progress and effectiveness in measurable terms will help in the early identification of things that "seemed like a good idea at the time."

4.4.5 Organizational Commitment and Participation

From the start, the central theme of an integration effort must be cooperation and focus on the corporate whole. A major barrier to the integration of corporate information systems is parochialism. In order to bring together disparate disciplines, a program is required with the following goals:

- o <u>Discovery by participants of their common ground</u>. This includes identifying common problems as well as emphasizing that all are contributing indispensably to the goals of the corporation.

- o <u>Discovery and appreciation of the problems that are unique to each discipline, as well as of the "mind set" that each discipline has developed over the years to assure its success</u>. For example, it can be difficult for MIS professionals to understand the environment of technologists in which data is "created from thin air." Conversely, although technologists understand the rigor required by their particular discipline, they are often not appreciative of the rigor required to handle large quantities of data due to the fact that electronic design and automated manufacturing is relatively new. Each group must learn from and accommodate each other. This applies from general categories such as business, engineering, and manufacturing, down to the specific functional areas within each of the categories.

A "public relations" program will be necessary to inform, educate, and solicit the cooperation of all impacted members of the organization. The goals of the program should be made clear. The importance of the program to management should be made equally clear. Incentives for compliance or participation should be considered.

Seminars and feed-back sessions should be held to identify problems, refine ideas, review suggestions, etc. Progress will be facilitated by cooperation, commitment and participation of the broad spectrum of people involved. Progress is difficult (though not impossible) in environments where the effects of integration are received as a necessary (or superfluous) evil.

4.4.6 A Note on Conflict Goals

Usually, the goals in an integration program are laid over original operational goals. This can often lead to conflicting goals. For example, a common goal for organizational units is to minimize cost for a given level of service. Occasionally, an integration program will cause one group to incur increased cost for what appears to be the same service. This increased cost is significantly offset by savings in other groups, or leads to improved reliability in a product or service.

If examined by itself, the increased cost represents a "failure" by the group's management. Examined in context, the group has increased its level of service, and therefore suffers increased expense.

4.5 METRICS OF INTEGRATION

4.5.1 Introduction

One of the major challenges of an integration project is determining its real value to the corporation. This challenge should be faced both at the inception of an integration concept and during the lifetime of any resultant projects.

A typical situation is that an integration project is proposed and defended in the hope that it will produce real benefits. If management decides to gamble on these projects, it is highly likely that the benefits (if any) will never be quantified. As more and more resources are poured into the project, hope typically wears thin and evidence of real progress is required by management for continuation of the effort. When management asks for evidence of progress, they usually want evidence of solving real business problems, not evidence of the adoption of new technologies.

Another important reason to be able to determine the value of integration is to manage expectations. User organizations need to know not only what their commitments are to the integration project, but also what benefits they can expect to accrue from the project. When users have unreasonable expectations, the integration project may be viewed as unsuccessful, even if it meets or exceeds the expectations of the information technology group facilitating the work.

The thesis of our panel was that value cannot be determined without metrics. We identified four major classes of metrics that can be used to assess the progress of an Integration project: data, systems, organizational, and cost/benefit. We focused on quantifiable, measurable parameters. The lists are undoubtedly incomplete, but represent a good start. Some of the metrics are binary: a factor contributing to integration that either exists or does not. Some are measurable in degree, others can yield absolute numbers.

4.5.2 Data Metrics

The following data metrics relate to the definition, storage/implementation, and management of data resources. The first three metrics are viewed as the most important. The prioritization of the others is optional.

o <u>The degree of data redundancy and overlap across data stores</u>. If a single baseline version of a data object exists, then there is evidence of integration progress. Integration is difficult with uncontrolled data replication. Which version of the data object is the correct one?

- Standards for data naming. The existence of a data-naming standard and the scope of its acceptance and enforcement within the corporation are important factors in enabling integration. Naming standards facilitate communication among developers and users; their adoption is necessary to achieving and promoting consistent understanding of data semantics.

- Existence and scope of a data dictionary and common data model. Data dictionaries are essential tools for managing data resources, integrated or not. Giving a dictionary knowledge of the corporation's common data model means that the dictionary can become an active component in integration. The common data model need not be (and probably will not be) developed and implemented on an enterprise-wide scale all at once. Rather, its scope should evolve through time. Start with a kernel data area that addresses visible problem areas for the business.

- Use of database management systems and databases instead of files. Because DBMSs promote data sharing (while protecting data from the perils of concurrent access and a variety of kinds of failures), the scope of their use is an indicator of progress in integration.

- Distinction between private, project, and enterprise data. Once an organization begins to distinguish between these types of data, then it can remove ownership of data from individuals' hands and put it in the enterprise's lap. Enterprise data is typically data that is accessible to integrated applications. Project data represents integration on a smaller scale.

- Use of a system development methodology that encourages integration. Such a system development methodology provides the policies and procedures, techniques and ideally tools that help to manage integration projects and to build integration skills into an information group. A systems development methodology that encourages integration usually emphasizes data modeling, model integration, and evolution of a common data model of enterprise data.

- Use of a common data model to guide application development. If each new application starts over with its own data definitions, integration will never occur. By contrast, evolving a common data model as applications are developed builds the glue that integrates applications. The common data model shows how the data items relate to each other, transcending the boundaries between individual applications.

- Mapping of common data model to implemented data stores. If the common data model is to play an active role in achieving integration, then the common data model must be mapped to implemented databases and files. Then the common data model can be used to guide application requests to their desired data, independent of the data's implementation structure or physical location.

- Granularity of data definition and management. Managing data at the file level typically indicates less integration than managing data at the entity level. The finer the level of granularity, then the more precise the definition of data semantics can be. Data semantics must be well understood and clearly communicated to achieve integration.

- Connectivity and transparency. The greater the span and scope of data stores that can be accessed from a common data interface, the higher the degree of integration. If an organization has a variety of kinds of databases or files that can only be accessed by separate specialized interfaces, then this data is not part of the integrated data resource.

- Penetration and automation of configuration management, consistent with responsibilities and accountabilities. The more pervasive configuration management is, the more integration has probably been achieved. Configuration management is important to control the release of information. Configuration management can be manual, but automated support is a good indicator of having achieved a degree of integration.

4.5.3 System Metrics

The system metrics relate to hardware, networks, operating systems, user interfaces, and applications. The most important metrics are the first two listed. The prioritization of the others is significant.

- Time and resources required to install and integrate new applications and systems. Integration does bring with it some overhead for data definition, data model integration, and so forth. However, a well-integrated system minimizes the overhead and reduces the effort required to bring new applications and systems onboard.

- Extent of heterogeneity. Systems with many heterogeneous components (hardware, operating systems, database management systems, and so forth) are more difficult to integrate than are more homogeneous systems. The extent to which heterogeneity is hidden from the user is a good indicator of progress in integration.

- Time and resources required to develop new applications. With a mature common data model, adding a new application typically becomes a matter of defining the logic of a new transaction using existing data and sometimes existing processes. For applications that use data known to the common data model, integration will reduce development resource requirements.

- Reusability of application code, data definitions, and data stores. The higher the degree of reusability, the greater the extent of integration. If every new application has to be developed from scratch, there is much room for improvement.

- Reusability of data, especially designs. One of the typical goals of integration in an engineering environment is to get greater leverage of engineers' work. This can be achieved in part by making designs reusable. This implies having a facility to catalog designs so that they can be found, usually based upon product definition characteristics. A high degree of reusability would be an indicator of having achieved some degree of integration.

- Completeness of product definition capture. Product definition is typically the result of efforts of teams of engineers, who may specialize in a variety of functional areas and disciplines. Complete and coherent product definition capture represents integration across these areas. Product definition data then is a facilitator of concurrent engineering, and can ease the interfaces between engineering and manufacturing.

- Consistency of user interfaces across related applications. A user ideally should interact with applications pertinent to his or her job in a unified, consistent manner. Having learned the user interface style for one application, the user should be able to anticipate (without additional training) how to interact with a related application. The more consistent the user interfaces are, the more integrated the system will appear.

- Maintenance costs of applications and systems. High maintenance costs may be an indicator of a myriad of non-integration linkages between system components. Integration should reduce the resources required to convert data from one file format to another, to maintain replicated data, and so forth.

- Extent of specification and control of system interfaces. A well-documented system architecture is key to effectively implementing integration technologies. The architecture should specify standards, interfaces, policies, and procedures, and should guide the development of the integration environment.

4.5.4 Organizational Metrics

There are several organizational metrics. The first two are the most important; the prioritization of the others is optional.

- Procedures for communicating and sharing data. With inadequate procedures for communicating and sharing data across functions and disciplines, there will be redundancy of work. Evidence of room for improvement in this area includes re-entry of data that already exist in computerized form, manual consolidation or reorganization of data from computer-generated reports, inconsistent data on reports at various levels of aggregation (as when the sum of several department totals do not match the sum of division line items for those departments), and so forth. Procedures for communicating and sharing data affect the way that people do their jobs.

- Level of explicit commitment for integration. The higher in the organization structure that stated commitment for integration is, the more likely that integration projects will yield real business results. Integration past a certain scope requires cross functional coordination, which must be supported by the management involved.

- Existence of goals and objectives related to integration. An organization that is committed to integration will have both personal and organizational goals and objectives related to integration. These goals and objectives, in aggregate, will help to ensure that individuals see how integration is an important part of their jobs and that all aspects of the integration project receive sufficient attention. These goals and objectives assign explicit responsibilities for integration.

- Performance appraisals tied to integration progress. A logical fallout of having personal and organizational goals and objectives related to integration is to make people accountable for achieving those goals and objectives. This can be done effectively by using integration progress as a measure in an individual's performance evaluation, guaranteeing that it will get attention! An organization's performance should also be evaluated based on integration goals and objectives, with progress made visible to appropriate management.

- Integration-related policies and directives. Directives (indicating what is to be done) and policies (indicating how to accomplish the directives) for integration need to be established and enforced. These directives and policies can help to ensure effective cross-functional communication and can help to make responsibility and accountability for integration explicit. The first directives and policies established are typically for Data Administration, for example, establishing change authorities for shared data.

- Training programs. People need to be trained in the concepts and procedures instituted as part of the integration project. If there are explicit training programs in place, with a high degree of participation by the appropriate people, then the integration project will have a relatively firm foundation on which to build. Again, integration affects people's jobs. If people do not do their jobs differently, nothing will change. Or, to turn the statement around, if people are not doing their jobs differently, nothing has changed, even if much money has been spent studying integration concepts.

4.5.5 Cost/Benefit Metrics

The cost/benefit metrics assess the real impact of an integration project on the corporation. They are all quantifiable, and all measure aspects of the business that contribute directly to the bottom line. They are all the kinds of metrics that should be considered in determining the objectives of an integration project. Depending on the scope of the project, some of the metrics may or may not be appropriate.

- Design cycle time. A major area where an integration project may have a positive impact is in reducing the cycle time required to design a product. The aspects of integration that help to reduce design cycle time are: being able to reuse design data, reducing design iterations by improving cross-functional communications, early weed-out of unattractive designs, and so forth.

- Manufacturing cycle time. Another major area where an integration project may have a positive impact is in reducing the cycle time required to manufacture a product. The aspects of integration that help to reduce manufacturing cycle time are: integrated design of the product and its required tooling or fittings, improving manufacturability of product designs by improving cross-functional communications, reducing manufacturing reworks, and so forth.

- Indirect (non-touch) labor. An integration project can have positive impact in reducing the headcount required to support direct design and manufacturing activities. Integration can make not only touch-labor more efficient, but can also make communication among management and other non-touch labor more effective and efficient. A positive indicator of integration progress is indirect headcount reduction.

- Time to market. Another positive indicator of integration progress is reducing the time from product conception to market introduction. Design and manufacturing cycle times are components of this measure. Time to market can be a corporation's differentiator, making or breaking its competitiveness.

- Resources spent on translators. This metric is more granular than the other cost/benefit metrics listed above. If resource requirements are decreasing to translate data from one form to another, for purposes of passing data from one application or system to another, then integration progress is being made. In a non-integrated computing environment, each introduction of a new application program is likely to also introduce the need for multiple translators, either to feed data to new programs or to pass its results on to other existing programs. Each time the data formats change in any of those programs, the pertinent translators must also be changed. These expenses are commonly buried in a category known as "maintenance." They are expenses that should dwindle as the integrated environment evolves.

4.6 END RESULTS OF INTEGRATION

In this section, we outline the overall framework for integration of technical and business data. We first state the assumption providing a model of integration, then we point out the end results of the integration process and the system considerations for achieving those results. Finally, we provide a prognosis of what is likely to emerge in the near future that will affect the possible end results of integration.

4.6.1 Assumption and a Model

Figure 4-1 shows the general environment of the management of business and technical data. It consists of a diverse set of users, a set of systems and stored databases in different forms.

The USERS include end users who need the data for their job functions, e.g., engineers, designers, analysts, managers, clerks. They also include the intermediary people who design and implement databases, programs or systems to be used by the end users, and finally policy makers including middle and top management. People in this last category depend heavily on the results of integration to provide them information in the right form, with the right content to help in decision making.

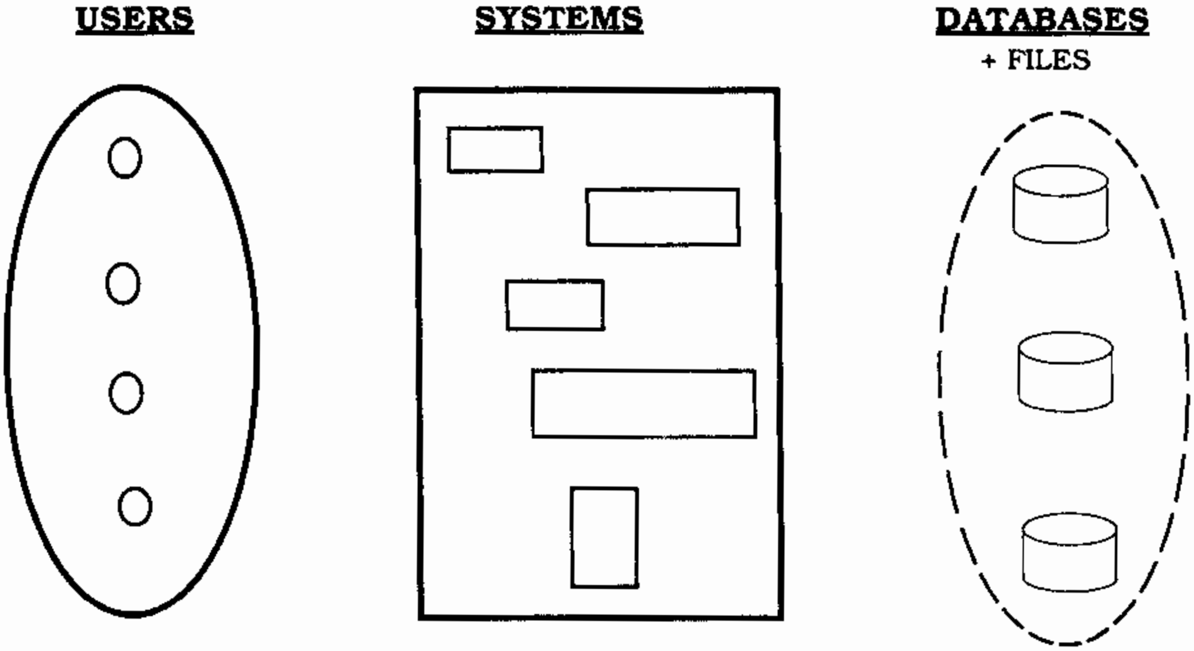


Figure 4-1

In the middle component called SYSTEMS we include hardware and software systems as well as organizational systems. The power and functionality of the hardware is continually improving. Moreover, hardware is the only area where the user is getting more computing power per dollar as technology evolves. The software developments are generally lagging both from the standpoint of meeting the end user needs and in terms of allowing the end user to exploit the available hardware powerfully. In the area of business and technical data, better and better user interfaces and software tools are expanding the range of people who can directly interface with the systems. These tools or software systems include spreadsheet packages, simulation programs, analysis packages for complex systems and drafting, and analysis CAD tools. Organizational systems include methodologies and procedures in place for collecting, organizing, manipulating, and disseminating business and technical information within organizations.

The final component in figure 4-1 refers to repositories of information in the form of FILES and DATABASES. They are on various media, including magnetic tapes of different kinds, disks in a variety of packaged forms, as well as archived hard copy information. We should actually include non-magnetically readable information in our discussion of integration since, historically, a large amount of such information still exists in that form. Some good examples are engineering drawings, product specifications, business correspondence, reports, manuals, etc.

When we discuss integration, we will assume the above scenario as outlined in figure 4-1.

4.6.2 Final Results of Integration

The results of integration can be viewed from different perspectives. They correspond directly to the three-component framework presented above.

- User Perspective. From the user's standpoint, when integration is accomplished, it results in a single user interface accessing a variety of system components, that in turn access diverse databases. To achieve the results, it is necessary to have a single conceptual model of data that can be used to capture the entire gamut of information. Current developments in semantic data models are addressing these diverse needs. Integration should also allow users to have their own "external views" of relevant information for each individual application.

- Database Perspective. From the standpoint of stored data, we can say that good integration is achieved when the following conditions are met:

  — In the stored form, each basic element of information is represented only once. This means redundancy is eliminated, making management of information easy. This has been the main goal of database management software systems.

  — Data should be integrated by removing the barriers between applications and between systems. Thus, within the limits of security, all potentially relevant data should be physically within the reach of a user or an application.

  — Databases must contain all possible cross-references or links allowing a user or an application program to navigate through all related data.

- Systems Perspective. If integration is achieved, the variety of systems should appear as a unified whole to user and applications. This results in a number of technical requirements that should be met by a well-integrated technical/business information system. It also raises several issues/problems. We address them in the next subsection.

4.6.3 System Considerations to Achieve the Results of Integration

In order to achieve the above results of integration, we must consider the requirements of computerized software/ hardware systems, both from a technical standpoint and an organizational standpoint. The technical aspect merits discussion because there has been a proliferation of hardware, operating systems, database management systems, and application packages over the last 25 years, and particularly in the 1980s. To integrate all business and technical data for those systems is indeed a major challenge of the 1990s.

At the same time, the user population has been constantly growing, due to the advent of easy-to-use systems and packages that have enabled the most naive and untrained users to access and process data directly. Consequently, a large number of fragmented databases have been created and are growing in an uncontrolled way. To integrate them presents a major organizational challenge.

We highlight below the technical consideration that must be dealt with in order to achieve integration:

- Need for new architectures. Today's software systems are developed for specific hardware under one or more fixed operating systems. Providing them a common interface, thus allowing a user to access and interrelate data among diverse systems, remains a research problem. Standards in interfaces, models, languages, etc., would alleviate the situation in which there is a loose-coupled integration of information processing systems involving repositories of data in the form of files and databases. "Loose coupling" allows substantial autonomous control, yet affords an integrated "global access."

- New for "common" building blocks. In the above types of architectures, there will be need for some common (possibly standardized) building blocks from which any system can be configured. These would include operating systems, programming environments, standard software components (such as the graphics core systems), common data models, standardized query languages, and even common windowing type facilities.

- Security specification and enforcement. Integration implies sharing. To control the sharing of programs and data, elaborate mechanisms for the specification and control of security would be needed.

  The challenge for future integrated systems spans the spectrum from providing conceptually integrated views and uniform user interfaces to dealing with the nitty-gritty details of physical data merging, redundancy control, efficient cross-referencing indexes, etc. Future system architectures would need to be flexible and open-ended to support a variety of levels of coupling and user-controlled parameters.

In order to achieve a desired level of integration of its business and technical data, an organization must cope with the following issues:

- Control of user population. Given certain security provisions, this deals with the control of authorized use of integrated information by users. The control problem becomes increasingly severe as we include more and more information in the integration.

- Organizational boundaries. Referring back to figure 4-1, the system components (hardware and software) as well as the data itself belong to different organizational units. Along with integration comes the problem of managing the systems and the data across these organizational unit boundaries.

- Training and education. Even with the easy-to-use systems of today, there is still a major problem of training the end users and the designers on every new system. The number of systems one has to learn has an adverse effect on the systems being fully utilized. Integration, if achieved properly, can cut down on the users' training requirements. Ideally, there should be a single application system and a desired database.

- Dealing with historical data, legacies, and inertia. Typically, users inherit systems and data and have a tendency to continue to use the same perpetually until they are discarded. With integration, it is necessary to accommodate all historical information into a single environment. "Old users" may severely oppose using the new integrated environment. This problem would need special managerial and interpersonal skills.

4.7 FUTURE ISSUES

The above discussion of the results of integration presumes the current technology and the visible trends. It has also implicitly considered only the components from figure 4-1. In terms of the future technological possibilities, the following issues will have to be addressed:

- Use of "intelligent" systems. In the scenario presented above, a predefined integrated conceptual schema is developed from which external user views are derived as needed for different applications. Another possible approach would be to consider "on-the-fly" ad hoc integration. In the latter, a user's request for information is processed by an "intelligent request processor" that determines what information systems and databases are needed by the user. It then proceeds to establish appropriate connections and uses the cross-referencing knowledge to retrieve and compile the necessary information. The FIB (federated information base) research project at the University of Florida under the direction of Prof. Navathe is researching this problem.

o Along the same lines as above, it may be desirable in the future to build "tailored" database management systems by using standard parameterized building blocks and system components to provide the best logical interfaces and physical design options for a given set of application needs. Experimental research is under way at IBM research, the University of Texas, and the University of Wisconsin in this area.

o Multimedia database management will soon become a commercial reality. We expect the results of integration to appear in the form of integrated modules containing text, voice, and images. This is likely to have a major impact on the way we keep documentation or write memos to disseminate business and technical information.

It is expected that as the technology progresses and more and more people start using computerized systems, the data subject to integration will increase in volume as well as in complexity. A number of challenges lie ahead in terms of technical aspects like interfaces, data models, languages, and operating systems as well as the organizational problems of security, controlled access, incorporation of historical information, etc.

5. THE INTEGRATION OF SYSTEMS PLANNING, DEVELOPMENT, AND MAINTENANCE TOOLS AND METHODS

John A. Zachman

CHAIRMAN

Biographical Sketch

John A. Zachman is a consultant for IBM's Applications Enabling Marketing Center. He joined the IBM Corporation in 1965 and has held various marketing-related positions in Chicago, New York, and Los Angeles. He has been involved with Strategic Information Planning methodologies since 1970, and has concentrated on Information Systems Architecture since 1984. In 1989 he joined the CASE Support organization of the Applications Enabling Marketing Center where he continues his work on Information Systems Architecture.

Mr. Zachman travels nationally and internationally, speaking and consulting in the areas of Information Systems Planning and Architecture, and has written a number of articles on these subjects. His current responsibilities include working internally with IBM as well as externally with IBM customers in supporting management with information systems.

PARTICIPANTS

Jean Berube
Bruce Buckelew
Donald C. Burnstine
Douglas Erickson
Dale Goodhue
James Hosmer
William Inmon
Beverley Kahn
Anne Miller
Linda Nadeau
Gerry O'Beirne
T. William Olle
Gerard Otten
Judith Quillard
Gary Schuldt
Charles Shoecraft
Richard Smith
Douglas Snyder
T. Travers Waltrip
Miles Welter

5.1 INTRODUCTION

The 1988 National Institute of Standards and Technology Workshop on Information Management Directions was comprised of five panels and focused on the subject of integration of information within the enterprise. By way of introduction, it is interesting to note that the dictionary definition of "integration" is: "the condition of being formed into a whole by the addition or combination of parts or elements."

Each of the five panels addressed a different aspect of "integration." The aspect that is covered in this panel report relates to the formation of the "whole" enterprise through the addition or combination of the various information systems "parts" or "elements" that are produced over the process of planning, developing, and maintaining the enterprise's information systems.

It is in the context of the "integrated" enterprise that the "integration" of information systems tools and methods is meaningful and in which it is possible to speculate about the future directions regarding the tools and methods.

The panel also discussed the sociological implications of "integration" in the enterprise, both from the perspective of the enterprise itself as well as from the perspective of the Information Systems organization within the enterprise.

The major conclusions drawn by the panel were that Information Systems planning, development and maintenance tools and methods must evolve dramatically to make integration cheaper and easier. Also, and maybe more significantly, the culture of the Information Systems organization as well as the enterprise must change drastically if the enterprise is to compete effectively in the dynamic environment so uniformly predicted by social and economic prognosticators.

The report summarizes and details the panel's findings.

5.2 APPROACH

5.2.1 Adoption of a Framework as the Basis for Discussion

The panel on Integration of Systems Planning, Development, and Maintenance Tools and Methods decided that it would be useful to adopt a framework to serve as a basis for discussion. This framework would pre-define a set of logical components and their relationships and, in so doing, minimize the time taken by a diverse group of Information Systems professionals to agree upon basic definitions and terminology. This would maximize the time available during the workshop to focus on the essence of the issue, "integration." The framework that was selected was one with which nearly all of the panel members were familiar, namely, the "Framework for Information Systems Architecture" [ZAC87].

In brief, the hypothesis that supports the adopted "Framework" is the following: In any discipline in which construction of complex engineering products is the primary objective, there is not a single architectural representation produced over the process of building the product, but there is a set of representations produced. These representations depict the perspectives of the primary participants in the design and construction process, namely the perspectives of the Owner, the Designer, and the Builder, as well as representations that establish the scope of the project and the out-of-context representations used for actual assembly and fabrication. For example, over the process of building a building, Architects and Contractors produce the following set of representations:

"Bubble Charts" (or sketches), depicting the gross size, shape, scope of the project, or a "ballpark" view of the end object.

Architect's Drawings, floor-plans, cut-aways, 3-dimensional representations depicting the Owner's view of the product.

Architect's Plans, engineering drawings and bills-of-material depicting the Designer's view of the product.

Contractor's Plans, technology and laws-of-nature constrained views depicting the "how-to-build-it" or Builder's perspective.

Shop Plans, out-of-context, detailed descriptions of parts or pieces (sub-components), the Sub-contractor's perspective.

Analogous representations can be found in Engineering and Manufacturing as well as in Information Systems, as shown in figure 5-1.

	Architecture/ Construction	Engineering/ Manufacturing	Information Systems
Ballpark View	Bubble-charts or Sketches	"Concepts Package"	Objectives, Scope Statement
Owner's View	Architect's Drawings	Work-breakdown Structure	Model of the Business
Designer's View	Architect's Plans	Engineering Design	Model of the Info. Sys.
Builder's View	Contractor's Plans	Mfg. Engineering Design	Model of the Technology (physical)
Out-of-context View	Shop Plans	Assembly & Fab. Drawings	Detailed Representations
End Product.	Building	e.g., Airplane	System

Figure 5-1

Further, disciplines that build complex engineering products also use different kinds of descriptions of the product in the design and construction process including, for example, descriptions of the material of the product (bills-of-material), descriptions of the function of the product (functional specs), and descriptions of the geometry of the product (drawings). The Information Systems analogues for these descriptions are shown in figure 5-2.

	Material	Function	Geometry
Architecture and/ or Manufacturing	Bills-of-Material	Functional Specs	Drawings
Information Systems	Data Models	Functional Models	Network Models

Figure 5-2

In summary, the combination of the two ideas that constitute the underlying logic of the "Framework for Information Systems Architecture" are:

- o There is a set of architectural representations produced over the process of building complex engineering products that represent the viewpoints of the Owner (the Model of the Business), the Designer (the Model of the Information System) and the Builder (Model of the Technology), and so on.

- o Different types of descriptions are used over the design and construction process including descriptions of the Material (Data), Function (Process), Geometry (Network), etc.

The combination of these two ideas produces the "Framework for Information Systems Architecture" as seen in figure 5-3. This "Framework" suggests that for each of the different types of descriptions, for example the Data Descriptions, there will be an Owner's View, a Designer's View, and a Builder's View, etc. Likewise, for the Function and Network Descriptions, there will be the Owners' Views, Designers' Views, Builders' Views, etc.

Therefore, in looking at an "Enterprise," this "Framework" puts some explicit specification around a set of Information Systems Models that potentially could be produced over the process of planning, developing, and maintaining information systems for the Enterprise.

The panel found the specification of these models useful in that it provided a "language" for quickly establishing meaningful dialogue between professionals of diverse backgrounds on the subject of integration.

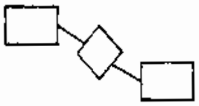
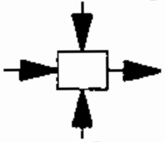
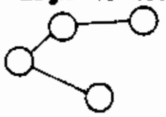
5.2.2 Definition of Integration

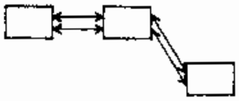
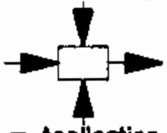
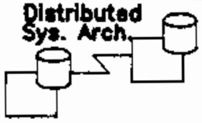
The "Framework for Information Systems Architecture" suggests by implication that integration could be defined in at least three different ways.

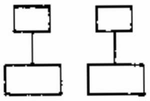
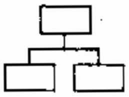
First, there could be "horizontal" integration, that is, integration of the cells (models) across a row of the "Framework," for example, integration of the Business Data Model (Entity/Relationship Diagram) with the Business Functional Model (Functional Flow Diagram) with the Business Network Model (Business Logistics System). "Integration" in this context would mean a "mapping" of one model to another as well as the maintenance of the mapping to ensure continuing consistency between the models.

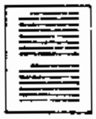
INFORMATION SYSTEMS ARCHITECTURE – A FRAMEWORK

	DATA	FUNCTION	NETWORK
OBJECTIVES/ SCOPE	List of Things Important to the business ENTITY = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in Which the Business Operates Node = Business Location

MODEL OF THE BUSINESS	e.g., "Ent/Rel Diag"
	
	Ent = Business Entity
	Reln = Business Rule
		e.g., "Funct Flow Diag"
		
		Proc = Bus Process
		I/O = Bus Resources (Including Info)
			e.g., Logistics Network
			
			Node = Business Unit
			Link = Business Relationship (Org, Product, Info)

MODEL OF THE INFORMATION SYSTEM	e.g., "Data Model"
	
	Ent = Data Entity
	Reln = Data Reln
		e.g., "Data Flow Diag"
		
		Proc = Application Function
		I/O = User Views (Set of Data Elements)
			e.g., Distributed Sys. Arch.
			
			Node = I/S Function (Processor, Storage, etc)
			Link = Line Char.

TECHNOLOGY MODEL	e.g., Data Design
	
	Ent = Segment/Row
	Reln = Pointer/Key
		e.g., "Structure Chart"
		
		Proc = Computer Function
		I/O = Screen/Device Formats
			e.g., System Arch
			
			Node = Hardware/Sys Software
			Link = Line Specifications

DETAILED REPRESEN-TATIONS	e.g., Data Design Description
	
	Ent = Fields
	Reln = Addresses
		e.g., "Program"
		
		Proc = Language Stmts
		I/O = Control Blocks
			e.g., Network Architecture
			
			Node = Addresses
			Link = Protocols

FUNCTIONING SYSTEM	e.g., DATA	e.g., FUNCTION	e.g., COMMUNICATIONS

Reprinted with permission from IBM Systems Journal, Vol. 26, No. 3

Figure 5-3

Similarly, horizontal integration would apply to the Information Systems Model Row as well as the Technology Model Row.

Second, there could be a "vertical" integration of the models down a column of the "Framework," for example, integration of the Business Data Model (Entity/Relationship Diagram) with the Information System Data Model (Data Model) with the Technology Data Model (Data Design). "Integration" in this context would mean the "transformation" of one model to the next, ensuring the assumptions implicit in higher level models are accommodated/reflected in lower level models.

Once again, vertical integration would also apply to the Functional Model Column and to the Network Model Column as well.

Third, there could be an integration of any one of the models in the "Framework" across the entire scope of the enterprise. For example, examining the cell in figure 5-3 that forms the intersection of the Business Model Row and the Data Model Column, integration could apply to producing an entity/relationship diagram (model) that would span the scope of the entire enterprise in a single, contiguous fashion. In this case it would constitute an "integrated," enterprise-wide model of the business data. Similarly, selecting once again any cell in the "Framework," integration could apply to producing a single, contiguous model that spans the scope of the entire enterprise.

In summary, "integration" could be construed to mean:

- o Horizontal Integration—integration of different kinds of models
- o Vertical Integration—integration of various points of view of the same kind of model
- o Intra-cellular Integration—integration of any given model, representing a single, contiguous view of the entire Enterprise from the perspective of a single cell.

Any discussion of integration would have to address at least these three possibilities, as their implications differ significantly.

5.2.3 Integration Questions

The questions that seemed to be logical to pose against each of the definitions of integration were:

- o What is the value, if any, for integration?
- o Assuming that there is value, what are the obstacles that keep us from integration?
- o What are the method and tool implications for integration?
- o What sociological impacts would such integration have on the business or on information systems?

These four integration questions were asked in relation to horizontal integration across the Business Model Row, the Information System Row and the Technology Row. They were also asked in relation to vertical integration down the Data Column, the Function Column and the Network Column. Last, they were asked in relation to integration within each cell in the "Framework" as it would depict a perspective on the entire enterprise.

5.2.4 Work Structure

The panel participants were carefully selected such that each of the nine major cells in the "Framework" had at least one panelist that was experienced, that is, <u>expert</u> in that cell. Therefore, the assembled panel represented a widely diverse group of professionals whose expertise covered every one of the nine major cells, or the broadest possible spectrum of Information Systems Planning, Development, and Maintenance for Data, Function, and Network and who, therefore, were well qualified to address all dimensions of the integration questions.

The panel divided up into sub-groups as follows:

To address "horizontal" integration:

Business Models Row	+Don Burnstine Doug Erickson Linda Nadeau	Judith Quillard Dick Smith	Doug Snyder Trav Waltrip

I/S Models Row	+Gary Schuldt Jean Berube Bruce Buckelew	Dale Goodhue Beverly Kahn Ann Miller	Gerard Otten Miles Welter
Techno-logy Row	+Bill Inmon Jim Hosmer	Gerry O'Beirne Bill Olle	Chuck Shoecraft

To address "Vertical" integration:

Data Column	Function Column	Network Column
+Linda Nadeau	+Jean Berube	+Chuck Shoecraft
Bill Inmon	Don Burnstine	Bruce Buckelew
Beverly Kahn	Doug Erickson	Dale Goodhue
Judith Quillard	Ann Miller	Jim Hosmer
Gary Schuldt	Bill Olle	Gerry O'Beirne
Dick Smith	Gerard Otten	Trav Waltrip
Miles Welter	Doug Snyder	

(The "+" symbol denotes the Chairperson for the group.)

To address intra-cellular integration, the same teams were used as those that addressed vertical integration, with the exception that the Chairpersons were assigned as follows:

+Judith Quillard   +Ann Miller   +Bruce Buckelew

The work products that were produced by each of the sub-groups comprise sections 5.4 through 5.12. Section 5.3 below presents some overall observations and conclusions that are derived, in a general sense, from those work products.

5.3 OBSERVATIONS AND CONCLUSIONS

5.3.1 Value and Obstacles Questions

First, with regard to the "value" and "obstacles" questions, a pattern appeared to develop along the lines of the three definitions of integration, as follows:

"Horizontal" Integration (i.e., Integration of the cells across the rows of the "Framework").

The recurring themes that appear to underlie the three working teams' perceptions of the values of horizontal integration are:

- o enabling the enterprise to manage change
- o providing Flexibility/Adaptability.

("Flexibility" was considered to relate to internal change, that is, change within the system. "Adaptability" relates to change of the enterprise as it relates to its external environment.)

The underlying explanation for the panel's observations regarding those values of horizontal integration probably relates to the concept of separation of independent variables. If independent variables are separated and managed independently from one another, then the probability that changes to one can be made without affecting the others appears to increase significantly. Further, if each of the independent variables is mapped to (that is, related to or "integrated with") the others, then changes to one that do impact another would likely be readily detected, evaluated and then could be implemented as appropriate.

The information systems implications of this would be that separation of the Data Models from the Functional Models from the Network Models, coupled with a mapping (or "integration") of the three different kinds of models with one another, would produce a systems environment in which change would be more readily manageable, that is, the system would be more "flexible," making the enterprise more "adaptable."

This appeared to apply consistently across all of the rows in the "Framework": the Business Model Row, the Information System Model Row and the Technology Model Row.

The obstacles to horizontal integration seemed to have underlying themes as well, namely:

- o No standard definitions and relationships exist for all the models in the "Framework" o "Inertia," that is, inertia of past practices and methods for managing the enterprise and its systems.

The substance of these obstacles apparently lies in the fact that, historically speaking, industry practice tends to embrace a "one-model-does-it-all" approach. That is, we have tended to attempt to imbed all variables (Data, Function, and Network as well as Business, Information Systems, and Technology) into a single "architecture." Therefore, for example, even though the data models are growing in acceptance vis-a-vis the functional models, little effort has been expended to date to become definitive about what belongs in which model, how the models relate, establishing standard definitions and relationships, etc.

The Network Models (at least the higher order models) have not had sufficient general usage (that is, they don't even tend to be produced in actual practice). Therefore, few generally accepted formalisms exist and little thinking has been done with regard to what belongs in the models and how they relate to other models. Regarding the "Systems Architecture" (Network Column, Technology Row), there is significant current focus, wrestling with vendor standards, international standards, etc. However, formalization of standards for the contents and relationships with other models has yet to emerge.

Because of the historic industry practice (inertia) of attempting to produce a single "architectural" representation to express multiple perspectives, there appears to be some reluctance to separate the independent variables. As a matter of fact, professional parochialism and/or vested interests even cause a strong inclination to imbed even more independent variables into single representations. Therefore the "champions" of the various design techniques, in attempting to "enhance" their products, present formidable obstacles to horizontal integration and thus inhibit the realization of the values associated with change management and flexibility/adaptability. The ultimate impact is to render the enterprise incapable of responding to the demands of its external environment.

Vertical Integration (i.e., Integration of the cells down the columns of the "Framework").

The uniform value pattern that arises from looking at integration of the Business Model with the Information

Systems Model with the Technology Model of each of the columns in the "Framework" appears to be: Quality.

An apparent explanation for this observation might be that the quality of any given model (cell) in a row is primarily dependent upon whether it was satisfactorily derived from the next higher order model (cell) in the column.

In general, three possibilities exist:

- If a higher order cell has not been produced prior to defining a lower order cell, assumptions about the content of the higher order cell would have to be made when producing the lower order cell. If the assumptions are in error, these errors will degrade the quality of the lower order cell.

- If a higher order cell is produced but not utilized when defining the lower order cell, then once again, assumptions would have to be made that could introduce error.

- If the transformation from the higher order cell to the lower order cell did not maintain semantic integrity or accurately carry over the constraints defined by the higher order cell, then error will be introduced into the lower order cell.

In each case, whether assumptions have to be made, or whether the transformation lacked integrity, the quality of the lower order cells (and therefore the quality of the resultant product) is degraded.

This would be consistent for integration of the cells of the Data Column, the Function Column, and the Network Column and supports the observation that the value for vertical integration relates to quality of the end product.

The obstacles to Vertical Integration are:

- inadequate transformation methods/tools

- lack of experience/skills

- strong user demands for implementations.

Underlying the obstacles is the inordinate user demand for "running code" and a perception in the data processing community that specification techniques (higher order models) are "mere" documentation. Therefore, insufficient time appears to be available to be taken either to build higher order models or even to acquire experience/skills for building higher order models. Further, there is not universal confidence that the methods or tools for performing the higher to lower order transformations are adequately defined to ensure a semantically (or otherwise) accurate transformation. Much more work needs to be done in this area to realize the values that vertical integration promises.

Note that without the ability to produce higher order models and accurately transform them into working systems, it is inevitable that errors will be introduced. Studies clearly show that the cost of rectifying errors in an implemented system is substantial, but beyond this cost is the cost of damage that the system errors can do to the enterprise itself.

Intra-cell Integration (i.e., integration of a given cell model across the scope of the enterprise).

The working groups' recurring theme regarding the value of integration within any given cell appears to be found in:

- o Reusability of existing design, and
- o Consistency/audit (exposing anomalies or inconsistencies, another source of errors.)

The key here seems to be seeing the total scope of potential implementations in order to identify (and therefore plan and design for) reusability. This would provide enormous potential for leveraging investments in development.

Although much industry attention has been focused on reuse of lower level components, reusable higher order models would provide even more avoidance of wasted project development dollars where projects have significant but subtle overlap.

The converse of leveraging reusability is identifying inconsistencies or anomalies. It is the inconsistencies and anomalies that are costly, not only from an operational standpoint, in that they require costly operational reconciliation, but also from a management standpoint in that policy inconsistencies (conflicts in objectives) cause resource dissipations. Even in the Network, inconsistencies/anomalies (called "incompatibilities" in the Network) require substantial investments either in hardware or software and on-going management to reconcile.

In any case, the value of enterprise-wide models appears to be reusability and/or identification of inconsistencies, which translates into increased productivity, in a positive sense by leveraging reusable "assets" and in a negative sense by reducing sub-optimization.

The obstacles to Intra-cell Integration appear to distill into several categories:

- o Existing investment—in applications, tools, technologies, methods, etc., (which perpetuates dis-integration)

- o Short term versus long term trade-offs (the cultural penchant for immediate gratification)

- o Perceptions that integration means "centralization" or "control" (raising the "ownership" question).

It is interesting to note that the promise of dramatically increased productivity through asset leverage (reusability) and the containment of resource dissipations from sub-optimization have not been perceived as sufficient to overcome the obstacles that constitute dis-integration. The questions that this raises are:

- o Is this because the methods and tools have not materialized to make intra-cell integration cheap or easy enough?

- o Or, is it because the potential has not been articulated or quantified?

- o Or, have we just not evolved to a stage where the impact of asset leverage and productivity take precedent over short term results?

In all likelihood, all of the above possibilities are operative.

Furthermore, the enormous existing investment in dis-integrated, implemented systems upon which an Enterprise might well be dependent for continuing operations not only is a deterrent in its own right, but spawns a host of vested interests that "philosophically" inhibit integration. This leads to the "ownership" question: who owns the implementation? The Enterprise? Or is it a component of the Enterprise? It is in this context that integration is perceived to equate to "centralization" or "control" and therefore constitutes an obstacle to achieving leverage and increased productivity.

5.3.2 Methods and Tools Question

After examining all of the answers to the methods and tools question from all of the working groups, it seemed appropriate to consolidate them into a conceptual structure of the factors that form the rationale or motivation for method and tool development.

In the context of Planning, Development, and Maintenance as suggested by the "Framework for Information Systems Architecture," there appear to be four conceptual areas of opportunity that could be addressed by methods and tools:

- Reduce "production" cycle time
- Facilitate change management
- Integrate existing systems
- Improve quality.

The specific method and tool directions identified by the working groups within each of the above categories are briefly summarized below.

Reduce "Production" Cycle Time

a. Normative Models

The concept here would be to "generate" a model (cell in the "Framework") based upon some external variables, reducing the time required to produce the model using a "discovery" process. The concept of normative modeling usually applies to models in the Business Model Row that establish the "Planning" context for downstream Information Systems design activity.

b. Automated Transforms/Mappings

Once again, the concept is to reduce the time it takes to produce a model (cell) by automatically deriving it from another model (cell). The terminology usage seems to be that a "transform" applies to deriving a lower order cell from a higher order cell; a "mapping" derives a cell from another cell in the same row.

c. Design and Documentation Aids

Here the idea is to reduce the time a designer needs by eliminating the paper of the design process, that is, designing directly onto electronic media (analogous to CAE or CAD in manufacturing) and generating paper (documentation) as required from the electronically stored models.

d. Pattern Recognition

Here the idea is leveraging reusability by identifying designs already completed that could be reused, avoiding the necessity to re-design from scratch.

<u>Facilitate Change Management</u>

a. Model Storage and Maintenance

This is the "storing" function in which each model (cell) would be stored independently from the others, such that changes could be made within a cell as long as they did not impact adjoining cells. Coupling the idea of versions with model storage would produce extensive possibilities for "configuration management" and change control.

b. Impact Analysis/Costing

When changes to a given cell impact the structure of an adjoining cell, it would be necessary to identify and cost those changes before allowing them to be made. If the storage facility contained the transform/mapping algorithms, not only could the impending changes be identified and costed, they could actually be implemented when authorized.

c. Automated Rule Management

Unlike Architecture/Construction or Engineering/Manufacturing, in Information Systems, the media for the actual product is the same as the media for the design of the product. Therefore, not only could changes be automatically enforced in the design process (as in a., Model Storage and Maintenance and b., Impact Analysis/Costing, above), but all the rules/constraints of the various design models could be automatically enforced during transaction processing if the design models were part of the transaction processing environment. The implication of this is that the higher order cells could be active in the sense that transactions could actually be processed by the design models and therefore all the design rules/constraints automatically enforced during operations.

d. Dynamic Process Path Selection

If the transactions were processed through the higher order models (as suggested in c., Automated Rule Management), orders of magnitude greater flexibility would be available for selection of process paths. The implications of this are that the business could dynamically restructure the transaction processing paths, only constrained by the business rules inherent in the data models, the functional modules in inventory and the installed network.

Integrate Existing Systems

a. Incremental Integration

The concept here is that as soon as any system (and actually, the logic applies to any model, or cell in the "Framework") is built, it becomes an "existing system" (or model). If it were not designed as integrated across the scope of the entire Enterprise (re: intra-cellular integration), then, as an existing system (or model), it is sub-optimal. Ultimately, like all sub-optimal systems, it would be a candidate for some type of post-integration at some point in time.

The tool idea would be to incrementally integrate the "pieces" of a "total" model as the pieces are constructed. Further, when the method/tool, in attempting to integrate a new piece with an existing piece, discovered that structural changes were required in the existing piece to effect the integration, it could invoke the change management facilities of tools like those described above in Model Storage and Maintenance and/or Impact Analysis/Costing.

b. Reverse Engineering

The issue addressed here relates to existing systems that are physically running in incompatible environments. In order to integrate them (or even to achieve the economics of converting them to compatible environments), they would have to be "reverse engineered." That is, higher order models would have to be abstracted out of lower order models until the technology independent model is achieved. The technology dependent model could then be forward engineered into a compatible technology environment. This concept, coupled with the concept of incremental integration (a., above) could produce compatible, enterprise-wide, integrated systems (models).

Note: The question here is whether it is, in fact, possible to logically deduce valid higher order abstractions that are only implicit in the "Technology Models." It is likely that it will <u>not</u> be possible in every case. This is particularly relevant in the Data Column. However, if rule-based processing approaches can be used to expose possible anomalies and assist the re-engineers in examining the implications of the assumptions, this obstacle may be less formidable.

c. Heterogenous, Distributed Systems

Here the issue is existing systems in incompatible environments in which there is no desirability or possibility for making the environments compatible. This would force the integration to take place operationally as the transactions are processed. The implication of this is the transaction would have to be processed through a higher order model, a technology independent model, in the sense of Automated Rule Management above, enabling the implementation to <u>appear</u> to be integrated even when the technology dependent designs (the physical environments) were not integrated, that is, were incompatible. The real question here is whether it is possible to produce a semantically consistent "Data Model" (Information Systems Row, Data Column) from more than one pre-designed "Data Design" (Technology Model Row, Data Column). (See ch. 6, the report from the Panel on "The Integration of Heterogenous Computing Environments.")

Improve Quality

a. Model Analysis

The concept here is to analyze any one of the models (cells) based upon some objective criteria that would define completion, or some other aspect of quality. This would include, for example, analyzing the highest order cells, in which planning decisions are made, based upon criteria that are designed to determine priorities, segmenting the models into implementable segments. The model analysis capability, coupled with the transform/mapping capability (see Automated Transforms/Mappings, above) would be very significant for controlling the quality of the end product.

b. Validation

The concept here is to provide the ability to analyze a piece of a cell to ensure that it is consistent with the rest of the cell, or to analyze a cell to ensure that it is consistent with the constraints of an adjoining cell. Both of these ideas contribute to end-product quality.

5.3.3 Sociological Questions

As in the methods/tools question, it seems useful to present the overall pattern of all working groups regarding sociological impacts of integration.

There are three major observations:

First, the concept of integration, taken to its logical conclusion, blurs the distinction between the enterprise and its information systems. The information systems are clearly a representation, and may be even the implementation of the infrastructure of the enterprise.

Second, integration would have substantial impact on the enterprise power structure in that it would change the concept of "ownership" and provide the flexibility to distribute power based on criteria other than merely "ownership." That is, if the old adage "knowledge is power" is true, then in the dis-integrated environment where systems are "owned" by departments and data is "private," power is the happenstance of "ownership." However, in an integrated environment where the systems are owned by the enterprise and the data is shared, power can be distributed on a rational basis independently of who has control of the data.

Third, integration would have an impact on the value system as well as the processes—of I/S as well as the Business—in that it would give either or both, great flexibility for dynamic infrastructure change. Therefore, integration is the basis for facilitating infrastructure and culture change in the enterprise as the external environment changes around it.

Within the purview of sociology, it seems significant to make the observation that integration and centralization are independent variables. Integration has to do with consistency. Centralization has to do with authority. It is true that in a centralized environment,· consistency can be legislated through exercise of authority. But it is equally true that in a decentralized environment, anarchy is prevented by decentralizing within the context of an integrated infrastructure.

The converse is also true. That is, an organization can be either centralized or decentralized on a dis-integrated base, although neither of these options is particularly effective. In any case, integration and centralization/ decentralization are independent variables.

This means that integration IS NOT tantamount to taking away the authority for independent action at any level of management. Integration IS providing all levels of management a consistent view of the data such that decisions could be taken on a rational basis. It also means that all levels of management would have consistent views of top management's strategies and objectives. This is all quite independent of where management chooses to grant authority for taking independent action.

It is useful to add that integration also makes change possible because, having explicitly described the rules of association of all the parts, the impact of changes can be readily understood and implemented when appropriate. In contrast, it is virtually impossible to change a dis-integrated or anarchistic environment, first because the impact of changes could not be discerned and second, as any one piece is changed, there is no forewarning of how any or all of the other pieces have to be changed to prevent discontinuity.

In summary, the perception that integration is centralization or control is no more valid than saying the chart of accounts prevents granting responsibility to spend money. In both cases, the consistent definition of the parts and the rules for their association have nothing to do with the delegation of authority for independent action. That is an independent variable.

Sections 5.4 through 5.12 present the work products produced by each sub-group of the panel.

5.4 HORIZONTAL INTEGRATION: BUSINESS MODEL ROW

5.4.1 Introduction

Why consider Inter-connection
Integration
Inter-dependence?

Re: Tom Peters "Thriving on Chaos"

Global Competition, Global Markets demand:

- o 400 times improvement in quality
- o Product cycles reduced to 38 weeks
- o Focus on quality and service

How do we make this happen? What role can I/S play? Is DIS-integration better for responding to change?

5.4.2 Value

- o Referring to the Business Models, "interdependent" seems to be a better term than "integration."
- o The business' nodes are interconnected for:
  - — Effectiveness
  - — Flexibility
  - — Adaptability
- o The business is interconnected or interdependent at some level. For example, a conglomerate, at the highest level, might only be interconnected financially.

- Interconnection provides the opportunity to support the reality of the business, for example, the interconnection of engineering and manufacturing.

- Interconnected/interdependent models permit and encourage iteration and "what if" kinds of question asking.

- Identifies reusable, redundant "chunks."

- When asking: should the business models be integrated? The immediate reaction is, "of course!"

- Considering inter-connected
  integrated
  inter-dependent definition and design
  versus <u>dis</u>-inter-connected
  <u>dis</u>-integrated
  <u>dis</u>-inter-dependent definition and design.

  Integrated definition and design:

  — supports organization behavior and structural change
  — enables control of the information resource
  — enables response of the enterprise to competitive thrust.

5.4.3 Obstacles

- General inability to translate the business vision statement into the information systems implications.

  — The vision statement spans the "columns"
  — Can't translate from Row 1 to Row 2

- Lack of business analysis skills (We lost the business analysts about 25 years ago—we converted them to programmers.)

  — Methodology
  — Tools
  — Models

- We are trying to use Row 3 tools and skills to do Row 1 and 2 kinds of things.

- The P.C.:
  - — Decentralizes control
  - — Offers _fast_, high quality (i.e., service) results
  - — _Dis_-integrates processes and/or data
  - — Later realizes that it is not isolated, and wants access to a database.
- Perception that "integration" implies "centralization."
- No tools for "Connectivity" (Network Column) models.

  Connectivity is key mechanism for interdependence. Without it, the Business:

  - — Can't optimize process
  - — Loses quality
  - — Loses control and tracking
  - — Loses synchronization.

5.4.4 Methods and Tools

- Need tools or methods within I/S for Row 1.
- Need ability to make a translation from Row 1 to Row 2.
- Need tool or method for documenting and controlling changes to the relationships between Data and/or Function and/or "Connectivity" (Network) and/or Business Practices/Rules.
- Need tools to produce inter-related models from Business "Vision," then to retain, maintain and control changes that affect them. Also, must deal with existing systems and new systems.

  (Normative Models are candidates)

5.4.5 Sociological Implications

- Defined and clarified models of information (processes and rules) changes the power structure.
  - — Information is power

— Impacts "old boy" network
— Trading information as a "barter" goes away
— Can audit to the rules.

- There is difference between the Manager (controller) of the Business Models versus the Administrator of the Business Models. (Where does the power reside?)

5.5 HORIZONTAL INTEGRATION: INFORMATION SYSTEM MODEL ROW

5.5.1 Characterization of the Information Systems Row

- Model of the information system

- Designer's view

- Technology independent

- Subset of the business model in the sense that this is where the automation boundaries are drawn.

5.5.2 Value

Integration defined as "consistency across all models in the row."

- Efficiency of the process, that is, the process of designing and building information systems. The further down the "Framework" you get, the more expensive it is to fix problems. Therefore, integration at the I/S model level makes the overall process more efficient.

- Consistency, completeness and flexibility of the product.

- Provide better input to the next level (the Technology Model Row).

5.5.3 Obstacles

- No accepted definitions of the components in the models (cells) and how they relate:

e.g.,	Entity	<——>	Data Store
	Attribute	<——>	Data Element
	etc.		

- o Lack of any accepted model at all in the "Distributed Systems Architecture" cell.

- o Lack of agreed upon mappings between columns (i.e., points of integration).

- o Practitioners that build cells are in separate organizations and don't talk to each other (i.e., Data Administration, Database Administration, Analysts, etc.).

- o Lack of a well-articulated "Owner's View."

- o Investment in existing architectures.

- o CASE tools are not integrated.

  Note that the tools affect how we think about our methods. CASE tools are to I/S what applications are to the business—and we are inviting the vendors to tell us how to run our business.

- o Too many choices—methods, tools, CASE products, vendors, educators, authors, etc.

  We need model primitives relative to each cell to "normalize" the proliferation of choices.

- o Methodologies are fragmented and support the "old" ways (which are column-oriented—i.e., not horizontally integrated).

- o Resistance to change within I/S.

- o Decentralization of I/S functions makes methods integration difficult.

- o Several hats worn by the same person makes rows difficult to differentiate (e.g., same person plans, designs, builds, tests, etc.).

- o No appreciation for payoff—the short-term view, particularly when it comes to funding.

- Time integration problem—migration from existing systems.

- Technology features determine model characteristics. (e.g., 20 years ago we thought data was hierarchical due to the existing database technology—IMS).

5.5.4 Methods and Tools

- We need to discover the basic primitives of components in each model (cell) (e.g., Process, Entity, Data Element, Data Group, etc).

  (These are the "atomic particles" out of which we build the models, systems, etc. Without them, we have no basis for "normalizing" the myriad of views of them.)

- Having discovered the basic primitives, we could require the vendors to conform.

- The "Framework" could become a "database" and the vendors could construct "views" (different visual representations of the models) from the primitives.

- The tools should be more graphic.

- There should be less redundancy in the primitive view.

5.5.5 Sociological Implications

- Analytic vs. heuristic approaches can influence the structure of the organization (e.g., end user views squeeze the rows together).

  "Analytic" refers to the classic "waterfall" approach to building systems.

  "Heuristic" refers to the iterative, prototyping approach, cycling through the models.

- Integration will restrict "freedom" but not necessarily creativity.

- The focus shifts to values and objectives of the models rather than organizations and procedures, how to do the work.

THERE IS NO CHOICE BUT TO INTEGRATE

- Once we understand the problems addressed by the "Framework," we can then allocate the functions to specific organizations.

  (That is, first we have to recognize that there is such a thing as a Data Model. Next we can understand the problems it is trying to address. Then we figure out who is to build it.)

- Development of the integrated "Framework":

  — Continuous innovation
  — Constrained by budget/projects
  — Need people to do methodological integration. (It's a project that needs users, etc.)

5.6 HORIZONTAL INTEGRATION TECHNOLOGY MODEL ROW

5.6.1 Value

- Consistency of data, processing

  — to the customer
  — to the systems developer
  — to the organization.

- "Seamlessness."

- Data/Process independence—separation of independent variables for purposes of managing change.

- Ability to work within the organization from different perspectives, not only from a project by project perspective.

  That is, to work "horizontally"—from the perspective of the corporation/enterprise and its concerns and interests as well as to work "vertically"—from the perspective of a project and its concerns and interests.

Note: Conflicting goals
Coordination difficulties

- Ability to blend the short and long term goals together in a seamless fashion.

5.6.2 Obstacles

(Note: these obstacles do not appear to be limited to the Technology Row alone, but appear to be universal for all rows.)

- Economies of scale
- Coordination
- Motivation
- Existing Systems
- Inertia
- Economics—this relates to the ability of an organization to cost-justify rebuilding a system purely for integration purposes.
- Integration is viewed statically when it should be viewed dynamically—integration is a "moving target."

5.6.3 Methods and Tools

- Method and tools tend to be more of a "columnar" consideration than a row "consideration," that is, "vertical" integration is more dependent on tools than is "row" integration.
- At the Technology Row level, it is clear that cross-referencing is required between the Data column and the Function column. That is, for every piece of data there must be a process structure to support it, and vice versa. Similarly, there needs to be a cross-referencing with the "Systems Architecture," the Technology Model of the Network Column.
- Cross-referencing in itself is not adequate—other factors are volumes, frequencies, etc.

- Regarding "distribution"—the implications of distribution are so pervasive that they affect everything else—therefore, distribution must be the first issue decided upon.

  Distribution decisions tend to be made based upon:

  — technology
  — economics
  — to some extent, business.

  Having established the distribution structure, the other technology models vary independently.

- The technology models are defined at the end of the design process and therefore quality and productivity are determined by re-iteration up and down the columns.

5.6.4 Sociological Implications

- If the wrong methodology/approach is chosen, people end up defending technology decisions rather than solving business problems.

- Data Modeling is, at times, used as a political device to shift control in the organization.

- Integration requires discipline—therefore the maturity of an organization determines the level of commitment it could have to integration.

5.6.5 Other Issues

- "Old" versus "New" integration.

  — We know what integrating "new" systems means.
  — What does integrating "old" systems mean?

- Operational versus decision support needs—there is a "Collision" problem between short program needs and long program needs—that ultimately leads to extract processing or separate databases.

- A data model, on its own is not adequate—but essential nonetheless.

- Cost justification at a MICRO level works against integration.

  Cost justification at a MACRO level works for integration.

  This conflict is very interesting—it explains part of the confusion within the organization. Managers with a "micro" view prioritize and act very differently than managers with a "macro" view. Reconciling the views is a real challenge.

- There is a real difference between external and internal integration

  — External integration is integration as perceived by the user of the system.
  — Internal integration is integration as perceived by the builder and maintainer of the system.

- The role and affect of "application generators" is not clear.

5.7 VERTICAL INTEGRATION: DATA COLUMN

5.7.1 What Belongs in the "Data" Column?

- The unit of analysis is a "thing" versus an "activity" (process). That is, if you can "do" it, it's a process and not data.

  (To most of us it's intuitive.)

- Business rule content

  Rules of Association ==> Data

  Rules of Transformation ==> Process

5.7.2 What is Integration? (Some Views)

- o Input/Output View:

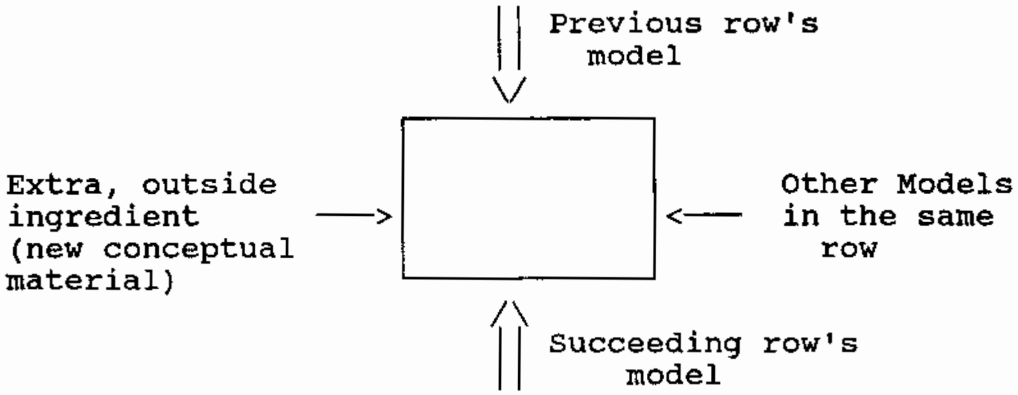

Previous row's model
Extra, outside ingredient (new conceptual material)
Other Models in the same row
Succeeding row's model

Figure 5-4

- o Traceability View (a.k.a. "impact")

  ("If x changes at a particular level, how do I make corresponding changes at lower/higher levels?")

  This relates to impact analysis, quality control —how do you recognize that you have reused conceptual material from above and brought new conceptual material in (or vice versa) to form a new model?

- o (A general point) Integration is independent of scope—scope is a project characteristic. Scope has to do with the "instances" of a column, not the "meta-content" of the column.

- o Integration down a column means one thing, integration across a row means another thing.

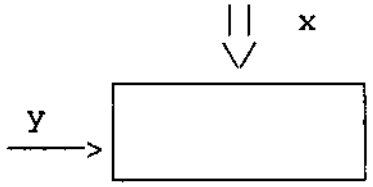
5.7.3 Content of Each of the Cells

Perspective	Data
Objectives "Ballpark" View	o List of critical business entities o Quick and dirty data model o Critical success factors
Model of the Business Owner's View	o Business Data Model o Analytic model of the business o Showing interrelationships (relationships) o Business rules (constraints, integrity, dependency) o Uniqueness (of entity) criteria o Quantitative (volumes of occurrences) o Business entity: descriptive attributes o Business purpose or role of an entity o Analysis of access control requirements
Model of the Information System Designer's View	Technology Independent o "Normalization" of business data model, including attributes o Business needs access model (logical navigation, process constraint) o Prescriptive model of an information system (data portion)
Technology Model Builder's View	Technology Dependent Data Model o Identify specific hardware/software products o Account for technology constraints & features o Factor in performance considerations o Security/privacy/recovery/backup o "Internal Model" (ANSI/SPARC) o Field sizes, etc.

Figure 5-5

5.7.4 Value

- Traceability of requirements through implementation.
- Avoids restarting from scratch—if higher level models are available, you don't have to re-gather the data required to produce lower level models.
- Validation and completion criteria are exposed at each level, which allows for traceability.
- Allows a change in business rules to be reliably propagated into the implementation system.
- Allows for impact analysis of change



e.g., If y changes and has an impact on the model, then it is possible to determine what changes have to be made to x (the next higher level model) to accommodate the change to y.

- Communication between rows (levels, perspectives). Each model is described from the perspective of the owner of the model, that is, the Owner describes the business, the Designer describes the Information System, the Builder describes the Technology, etc., and the models make it possible for communication in that they provide for shared vocabulary as the transformation is made from level to level.

5.7.5 Obstacles

- Lack of methods for transformation from one level to the next.

  — performing the integration
  — controlling the integration
  — evaluating the integration.

- Different perceptions of the various row roles. (For example, the business modeler who does not understand information systems design will have a difficult time negotiating with the designer. The information systems designer who does not understand technology will have a difficult time negotiating with the builder, etc.).

- Organizational commitment is difficult to get (power, resources, beliefs, etc.).

- Integration is perceived as (and may be) an impediment to flexibility (e.g., in some industries, the "cowboy" style of management seems to be in fashion and planning is out of fashion).

- Integration is perceived as centralization.

- Bottom-line/short-term payoff orientation.

  Business planners, for example, don't understand how models can be of value to them. Therefore they are reluctant to invest the time or resources required to build them.

- Perceptions at the higher level are less data-oriented.

  Higher levels of management find it difficult to talk about data without talking about how the data is used. Data design requires knowledge about data structure, but when you talk about semantics the idea of processing starts to creep in and it becomes hard to draw the line between the structure and the process which complicates building a data model. The concept of data as data seems to be lacking.

- Data is not thought of as a corporate asset but as an information systems asset (no matter how much the words "data is a corporate asset" are used). This is probably due to the fact that today, data is often not available to the corporate planners/strategists. They clearly are not presently using data models or meta-models in their day-to-day activities.

- Data analysts/business analysts misperceiving their role as "designers" (i.e., they tend to get prematurely physical, talking about physical structure issues, navigation issues, etc., when they should be talking about the underlying business rules.

5.7.6 Methods and Tools

Note: The method is the fundamental issue—the "tool" is "just" an implementation of the method.

(There was not universal agreement in the sub-panel on this issue. The disagreement centered around the word "just.")

- o Need methods that carry through all rows of the Framework—methodologies that cover more of the "life-cycle"—where the definition of "life-cycle" is much broader than current usage, for example, life-cycle that emphasizes reuseability, or life-cycle as related to an "entity," etc.

- o Need methods and tools that help us validate from one row to another (e.g., from analysis to design, etc.).

- o Need tools that will generate first-cut models.

- o Methods must provide for carrying business rules, constraints, etc., through the levels—not only carrying the structure (i.e., we need better information along the interfaces).

- o Need intelligent, expert, sensitive user interfaces to the tools that help the "user" do the job better. For example, interfaces that:

  - — help the analyst ask business questions
  - — help the designer cover all design bases
  - — provide tutorial help for novices and "short-cuts" for experienced users
  - — etc.

- o Need improvement in methods/methodologies That is:

  - — more options and paths through the methods/tools, not merely a "black box" type method where the intermediate models are not discernible
  - — better understanding of the purpose or steps of the methods—methodological primitives—to produce "recombinant" methods
  - — better criteria for evaluating evaluation criteria.

5.7.7 Sociological Implications

- o If I/S were indeed "integrated," it would be integrated into the business—that is, it would be seen as part of the business rather than an issue to be dealt with.

- o Methods integration requires "I/S Business" integration.

  (Decentralization, that is, dis-integration, is not a "black or white" issue. Different factors come to bear at different points in time, and therefore decentralization is a trade-off that needs to be understood and controlled.)

- o Data integration requires business integration.

- o Attempts to integrate will expose inconsistencies/incompatibilities that create human insecurities and power issues.

- o Integration makes more information available to more people, which could cause a given business person to do/perceive his or her job differently.

- o Integrated methods produce <u>metadata</u> that will result in integrated <u>data</u>. This could have significant ramifications, for example:

  — it would make security/privacy issues far more complicated
  — it would make the power issue more difficult to manage
  — it could cause information overload
  — etc.

- o Access to metadata—business rules would be modeled, recorded, and possibly publicized, which could have unpredictable impact on business units.

  (This problem is different from managing access to operational data.)

5.8 VERTICAL INTEGRATION: FUNCTION COLUMN

5.8.1 Function Column Definitions

```
REAL WORLD

Level 2—Business Model
   Responses to events
   Activity oriented                 (Abstract/concrete
   Time/material/products             Low level/high level)
   Resource, product life-cycle
   Precedence
Level 3—Information Systems Model
 (Technology Independent)
   Subset of level 2 ?
   Automation boundary <──────┐
Level 4—Technology Model      │... Same ?
 (Technology Dependent)       │
   Technology decision <──────┘
   Manual systems
Level 5—Specification
Level 6—Product
   Main part of what is delivered

REAL WORLD
```

5.8.2 Program of Work

- o Answer integration questions

- o Review interfaces between cells

  (Ignore the contents of the cells and concentrate on what goes on between the cells.)

5.8.3 Value

- o Enable consistency and requirements tracking at every interface

- o Enable quality assurance

  — Check point
  — Quality criteria
  — Product identification

- Enable capture of metrics along the value chain (i.e., from cell to cell)

- Increase of the value/cost ratio is directly linked to communication.

  (If the interfaces between the cells are not clearly defined, then a major expenditure is made in communicating about the contents of the cells. That is, we work at the boundary, either understanding or ignoring the work in the previous cell, and don't do the real work inside the cell.)

- User acceptance will be better if we implement integrated systems (i.e., integrate manual portion and automated portions) from an integrated life-cycle. In this event, there would be a higher probability of matching the implementation with the business function.

5.8.4 Obstacles

- No language to convey the whole problem to more than one person.

- Lack of communication requires 60% re-work.

- IS/IT reluctance to comply with user requirements, given restricted time frames.

  (As time grows short, we forget about doing what the user wants.)

- Organization boundaries may not allow the resources of all partners to be made available in sync.

  (In an integrated life-cycle, resources may not be available when you need them. For example, you can't hire a temporary user.)

- The whole process is, at present, people dependent—that is, the integration (whatever there is of it) is being performed by people.

- The user's perception of the process is that it is a "black box." That is, there is no level differentiation—which refers back to the resource availability issue—the users are not available except at the beginning and the end.

- Lack of skills in:

  - conflict resolution
  - communication
  - analysis versus development.

- Accounting systems cannot cope with missing values for cost/benefit in rows 1 - 3.

- North American short term view creates problems for high front-end cost, delayed return situation.

- Reusability is limited by our own experience and history.

  (We would much rather create new things than reuse old things.)

- "Discover the solution" mentality (ignore requirements).

5.8.5 Methods and Tools

- Partnership of IS/IT and the business—the result will be that the business will become:

  - less people dependent
  - more "process" dependent.

- Need multiple level of cost estimating/planning to secure execution of 2 steps at a time.

  (If commitment cannot be secured for the entire implementation, at least two steps are required to have the funding to do the proper job at the proper time.)

- Need methods/tools to produce short term, "quick fix"

  - to gain credibility, but also
  - for integrating quick fixes into the long-term portfolio.

- Need integrated tools (tools that "talk" to one another) to enable fast production for prototyping/ gathering right requirements.

  (The idea is to build the "wrong" system very fast and then build the "right" system. Without integrated tools, it is impossible to iterate through the development process in a reasonable amount of time.)

- Without tools, people concentrate on precision, not on judgement or direction.

- JAD-type approaches to integrate participants all along the cycle.

- Usage of normative models/reusable components built into the methodologies.

- Need to focus on innovation in the business, not innovation in techniques/methods.

- Tool integration depends on method integration.

5.8.6 Sociological Implications

- How to gain commitment from management/users when you are at the beginning of the cycle.

  (We are selling "futures" and we tend to sell things like entities and relationships. Maybe that is the wrong thing to try to sell. Maybe we should gather some information, make something run, and then sell the system.)

- User resistance to change needs to be addressed throughout the cycle.

- Loss of "ownership" (of a cell) versus integration.

  (When there is no integration, there is no problem finding owners. In fact the problem is too many owners. However, if there is good integration, it is hard to find an owner.)

- Sharing of the business purpose all along the column is the key to quality.

- Blurring of distinction between the business and IS.
- Creates a need for new skills—e.g., business analysts.

5.9 VERTICAL INTEGRATION: NETWORK COLUMN

5.9.1 Introduction

- Understand the content of the column.

  Technology resides in this column—and is a major issue. (For example, it could be observed that if there were sufficient, raw compute power to provide access between any one piece of data and any other piece of data, our interest in data models would be much less intense. Therefore, even though things can be built without understanding the underlying technology, we still carry around with us a collective knowledge of what can and can't be done with current technology. This limits, in almost every case, the scope of what we do.)

- Suggested alternate names for the column.

  There is a certain discomfort with the name of the column, "Network". It is an enterprise issue that needs an enterprise perspective, and "network" has a connotation that is somewhat limiting. Several alternative names were discussed, including:

  — Technology Infrastructure
  — Enabling Technology
  — Connectivity.

  However, no consensus was reached and so, for the time being, the sub-panel settled for "Network."

  The enterprise is made up of things that, once again, were hard to name. That is, "nodes" or "chunks," which are concentrations of activity—and the relationship between the "chunks." The "chunks" are aggregations of data and processes built around "where the decisions are made," not necessarily around the geographical dispersement. However, there may be a requirement for physical means of communication between geographical locations in order to make the decision process viable.

This raises the issue that there clearly is a relationship between the three columns, Data, Function, and Network (chunk), which may not be only three dimensional, but n-dimensional, as other factors, like organization (politics), may also be related. Further, decisions made in higher rows have dramatic impact on lower rows, and decisions about "dispersement" in higher rows have to be made early on because they become very specifically and physically geographic in the Technology Model Row. If the decisions are not made early on, then the Technology Model decisions of the Network column will constrain the logical or conceptual model decisions of higher rows by default.

o Integration of "connectivity" implies an enterprise with a hierarchy of decisions.

5.9.2 Value

(Or, cost of not integrating.)

o If there is no integration (from the top, down the column), then there will be DIS-integration, driven by the marketplace (what the capability of the technology is in the marketplace) and not by the enterprise purposes. For example, the existence in the marketplace of personal computers that can be purchased in large quantities by departments without having to obtain corporate authorization to do so, will result in systems all over the enterprise with their own data models, and with no hope of ever integrating them.

Two possibilities:

— A service approach, in which you allow people to move at their own pace until they make a request that takes them a step further. Then you must be technically prepared to deal with the technology as it comes to your door. If you are not technically prepared, then you risk the users concluding that I/S is stubborn and doesn't want to change.

(If you can't deal with complexity, then simplify.)

— Create an ability to forecast the future and participate in that process on an on-going basis in order to avoid failure in the future.

(Note: The OSI model is NOT an analog of the Framework. In the OSI model, you can ignore the lower levels, but in the Framework, you CANNOT ignore the lower levels.)

5.9.3 Obstacles

- Technical competence of the information organization.

- Ability to forecast availability of enabling technologies.

- The industry has become so specialized that the specialized knowledge tracks create boundaries that are difficult to integrate because the specialists can't think across the boundaries. What is needed is education to help future I/S managers view the whole and have some hope of understanding it. By the same token, future business managers must be educated to view the whole and understand the role and responsibilities of I/S.

5.9.4 Methods and Tools

- A primary tool is education—we need an Information Management Degree.

- The Framework needs to be viewed holistically rather than as parallel, independent processes.

5.9.5 Sociological Implications

- It is important that the responsibility for Information Architecture not reside in the I/S organization as long as the I/S organization does not understand that the objective is to serve the enterprise, not to optimize the technology.

- Current trends/capabilities in I/S technology allow end users to have enough technical expertise to create conflict with the I/S Department.

- The I/S organization must have service-oriented consultants who have a solid technological basis of understanding of the I/S Architecture.

5.10 ENTERPRISE-WIDE INTEGRATION: DATA CELLS

The approach the group used was to make general observations that applied to all cells in the column and then to take a specific cell to validate the general observations while noting any additional observations relevant to that specific cell.

5.10.1 Dimensions of Integration

- Addressing integration within a cell, it was important to distinguish between:

  a priori integration (systems designed with integration in mind) vs. post hoc integration (after-the-fact integration)

  Note: Most of our discussion during the workshop centered around a priori integration, whereas the real world problems were probably post hoc integration problems.

- Different issues must be addressed when discussing:

  integration of operational data vs. integration of informational data

  Note: We really didn't spend time on this distinction, however focused more on the operational data issues.

- Three dimensions of scope-related integration:

  - across business units
  - across business activities
  - across locations (this is the domain of the network column).

- Another consideration is at what level does the integration start? The assumption that was made is that given any cell, all cells above were integrated to the same scope. In practice, this may or may not be the case, since scope of integration of a lower level cell may be narrower than that of a higher level cell.

5.10.2 General Comments

- Each cell has its own tool/method implications

  - due to model characteristics, primitives, etc.

- Sociological implications—similar to previous observations

  - Levels 1 and 2—very similar
  - Level 3 (Conceptual Data Model)—very different
  - Level 4—very similar.

5.10.3 Value

Note: What we are trying to integrate are Business Data Models

e.g.:	A	B	C
	Engineering	Manufacturing	Finance

- The values of integration:

  - Communication between systems is possible. (Identifies interfaces)
  - Identifies inconsistencies/inefficiencies between systems. (Identifies anomalies—due to the lack of a common business language between the different systems)
  - Expedites development activities due to reusability of components of the business data model. (Identifies redundancies, identifies non-value-added tasks)
  - Multiple perspectives can be represented. (Across business units within a single company, across multiple companies within a holding company)

- May improve quality and save time in requirements process due to reusability of parts of the data model. (Data inventory—data item definition)
- Expose anomalies in business rules. (Produces a more robust data model, enables an impact analysis down the column)

5.10.4 Obstacles

- Different business units use different terminology and data names.
  - Homonyms/synonyms have to be resolved
  - Hard to detect commonality, conflicts, complementariness due to lack of robust, precise descriptions (no standard language for describing entities).
- Ownership (of data models) and "not invented here" syndrome.
- Lack of funds for integration. (Long-term vs. short-term payoff.)
- Difficult to articulate, in "dollar" terms, the benefits to the business.

  (High level personnel are less data oriented and more activity/function oriented.)

Note: Many of the group took issue with this observation. However, there was agreement regarding the difficulty in articulating the benefits. In this regard, it was suggested that a more viable approach may be to articulate the costs of <u>not</u> integrating.

- Who resolves conflicts?
  - Power to do so
  - Ownership issues
  - Business rule resolution.
- Conflicting organization/political goals. (In some cases it may be perceived as not in an individual's best interest to integrate the data.)

- Current models may not be sufficiently rigorous semantically to be integrated.

5.10.5 Methods and Tools

Note: The objective of the methods/tools is to merge business data models. That is, to synthesize/rationalize models, to identify/remove redundancy, to identify commonalities.

- High level model description required—meta model level with instantiations
- Need semantic models
- Need procedures/rules for identification and resolution of commonalities
  - Synonyms
  - Homonyms
  - Complementarity
- Some existing data modeling tools/methods do not lend themselves to integration (they emphasize structure rather than rules).

5.10.6 Sociological Implications

- Most of the obstacles identified have sociological implications.
- When systems are integrated (e.g., A + B —> AB) it forces us to look at the business in a new light—to identify new business problems and inconsistencies.
- The person who delivers the message (above) runs the risk of being "shot," or even worse, ignored.

5.10.7 Re: "Database Design Model"

Note: The Data Design Cell (Data Column, Technology Row) was chosen to validate the above general observations and to identify where potential anomalies occur.

Assumption: Integration has already been done at all higher rows.

- o Value of integration—same as above
- o Obstacles (basically consistent with the above)
  - — Incompatibilities among DBMS technologies
  - — Data naming conventions
  - — Conflicting performance tuning
  - — Conflicting denormalization
  - — Conflicting organization/political goals (DBAs)
  - — Who resolves the conflicts?
- o Methods/tools—Every cell really has its own method/ tool implications because of the different model characteristics and units of analysis.
- o Sociological Implications—Same as above.

5.11 INTRA-CELL INTEGRATION: PROCESS CELLS

The approach the group used was to select one cell, the Information System Cell, and as each integration question was addressed, to identify any additional observations for other cells.

5.11.1 Value

- o Reduced cost, improved quality.

  (Because, looking at the entire enterprise, we can predict results and impact of the relationships between processes. That is, we can see each process in context.)

- o More consistent results.

  (Due to dependency on common methodologies rather than on different individuals.)

- o Provides consistency in definition, acquisition and compensation of skill levels.

  (Other Cells)

- Value increases as we move from level 1 through level 5 (i.e., the models become less subjective and more explicit, and the cost of change increases).

5.11.2 Obstacles

- Conflicting and competing tools and methodologies.
- "Sunk" resources in existing inventory utilizing previous methodologies and tools.
- Continuing investment to maintain "old" and "new" (e.g., Tool/Method support, education).
- Tradition and history ("Last idea didn't work," etc.).
- Missing definition of product of cell and its associated evaluation criteria.
- Competition between creativity and conformance.
- Cost of retrofitting of "new" (method) to "old" (product).
- Process can be expressed as data or vice versa (e.g., derived data).

5.11.3 Methods and Tools

- Tools will need to derive different representations from the same information.
- If more than one tool/method is required, the tools/ methods must comprise an integrated (no overlap) family.
- Must be adaptable to size (i.e., tailor cost of use to project size).
- Maintain version control.
- Impact analysis between current and proposed versions.

5.11.4 Sociological Implications

- o Tradition and History ("last idea didn't work").
- o Competition between creativity and conformance.
- o Provides consistency in definition, acquisition and compensation of skill levels.

5.12 INTRA-CELL INTEGRATION: NETWORK CELLS

5.12.1 Introduction

o AKA: Node - Line - Node
Infrastructure
Geography (Schematic and real)
Technology Architecture
Delivery Mechanism

o Scope

Hardware and software that support the storage (node), processing (node) and distribution (or communications or transmission) (line) of information from corporate centers to workstations.

(NOT the storage, processing, distribution of physical goods).

Node = All pieces of hardware and systems software involved with the delivery mechanism.

Line = transmission or distribution mechanisms.

o Purpose

To deliver functions and data to users in a timely, reliable and cost effective manner.

o Some general characteristics

- — Flows from the culture of the business.
- — Supports the business by delivering the data and functions of the other two columns.
- — Supports (directly) the other two columns by providing the architectures, products and configurations that will allow them to work.

- Constrains the other two columns by limiting the architectures and products that can be used and by providing (or not providing) technical feasibility of proposed data and functions.
- "Prices" the infrastructure required to deliver the data/function required. (Residency, transport and "crunch" all have costs associated with them.)
- Always works best when it works in conjunction with the other two columns.

5.12.2 Some Characteristics of the Cells by Level

LEVEL 0

Vision of the Business

- Mission (products and markets)
- Business Environment (competitors, substitutes, finances, government regulations and other factors that have an impact on business operations)
- Culture
  - Centralization vs. Decentralization
    Decision making
    Management processes
    Operational control
  - Formal vs. Informal

  (These are very important to understand so that they can be incorporated in any design considerations.)
- Geographic Boundaries

For all of the above

What is it now?

What is the vision of the future?

What are the business strategies and plans to get there?

LEVEL 1

"Ballpark"

What is NEEDED by the cell:

- o A basic understanding that there is a need to control the integrity of the infrastructure

- o What products/types of products are offered by the enterprise

- o Where (by location) will different functions be performed, including a rough "flow,"
  e.g., Plants in Little Rock, ...
  Data Center in Canton, ...
  Home Office in San Francisco, etc.

- o What "types" (in general terms) of data are required to support the functions

What is PROVIDED by the cell:

- o List of the things involved, i.e., scope of the infrastructure in order to

  - — estimate or measure <u>gross</u> volumes
  - — estimate or measure <u>gross</u> performance requirements
  - — "configure," i.e., <u>grossly</u> size process, storage and communications requirements
  - — determine <u>gross</u> feasibility—price (?)

What is ESTABLISHED by the cell:

- o The highest level principles from which the architecture will be derived.

  For example:

  - — it will be simple vs. functionally rich
  - — it will be common or standard vs. diverse
  - — it will be medium risk vs. high risk
    etc.

LEVEL 2

Model of the Business
(Owner's View, "Logistics Network")

What is NEEDED by the cell:

(Where possible)

- o Volumes (that can be translated into rough process/storage/distribution numbers)
- o Performance characteristics
- o "New" technologies envisioned, if any
- o Ownership of infrastructure architecture

What is PROVIDED by the cell (to the function and data columns):

- o Technical feasibility (yes/no)
- o Cost vs. performance trade-offs
- o Integration feasibility on existing infrastructure

What is ESTABLISHED by the cell:

- o Current information infrastructure
  - — High level
  - — Pictures
  - — Words
  - — Principles

LEVEL 3

Model of the Information System
(Designer's View, "System Architecture")

What is NEEDED by the cell:

- o Understanding of I/S technologies available at architecture level
- o Any new functional requirements required for design

- Volumes and performance requirements from function and data columns (next refinement)

What is PROVIDED by the column:

- Specific "architectures" that are approved to support the functions and data
- Assistance in the selection and use of architectures if more than one choice exists
- Price estimates
- General configuration

What is ESTABLISHED by the cell:

- Explicitly stated system principles
- List of supported "architectures"/vendors
- Usage guidelines

ISSUE:

- Which column owns security?

LEVEL 4

Model of the Technology
(Builder's View, "System/Product Architecture")

What is NEEDED by the cell:

- Industry/Product knowledge
- Specific volumes and dates
- Compliance to architecture and product standards by data and process people—inside and outside the I/S organization

What is PROVIDED by the cell:

- Assistance in use of approved products

- Implementation/integration plans of network components to support data/function
- Capacity plan

What is ESTABLISHED by the cell:

- Specific list of products with configuration and usage guidelines that will work, are supported, and fit within our architectural principles
- Either acquisition authority or the right to exclude from infrastructure

LEVEL 5

Detailed Representations (Out-of-Context View, "Network Architecture")

What is NEEDED by the cell:

- Programs and databases that
  - Meet user requirements
  - Work
  - Comply with network architectures

What is PROVIDED by the cell:

- Infrastructure upon which
  - Data will reside
  - Programs will run
  - Users will have access
- Feedback/actual cost

What is ESTABLISHED by the cell:

- Physical capacity
- Management processes required to meet service levels.

LEVEL 6

Functioning System

What is NEEDED by the cell:

- o Ability to ensure data integrity
- o Programs be maintained.

What is PROVIDED by the cell:

- o Operations of infrastructure where appropriate
- o Support (infrastructure) to other "operators"/user-operators

What is ESTABLISHED by the cell:

- o Dollars and resources

5.12.3 Issues

- o These are examples but not industry accepted models and conventions.
- o Data and function columns are dispersing, and not generally owned by I/S professionals—process and data are DISintegrating.
- o Network "DISintegration" has occurred to varying levels. Therefore, "network" architecture will be more or less difficult to implement depending upon the state of the mess.
- o If not understood and endorsed and supported at the highest level in the organization and at the highest level of the architecture, it won't work.
- o It must be (and be perceived as being) ENABLING, not DISABLING.

5.12.4 Value

- o There is NO value to supporting interfaces that didn't have to exist to begin with

- — They are expensive
- — They are specialized
- — Long term, they won't work

- o Long term survivability of the infrastructure

- o Must be the combination of "central plan" and market determined

5.12.5 Obstacles

- o Existing physical inventory

- o "Belief" that architecting the infrastructure stifles creativity

- o Existing data/programs with no funding or ability to replace, phase out or kill, built before network architecture

- o Lack of ability to sell importance to top executives

- o Everyone is a technical "know-it-all"

5.12.6 Methods and Tools

- o Few exist

- o Need diagrammatic language/conventions to depict the complexity of the network environment

  - — Re: GUIDE project on Software Planning
  - — Re: System Planning Grid—IBM Systems Journal Summer '87
  - — etc.

  Note: We have Entity/Relationship diagrams for data, data flow diagrams for function—we need "node/line diagrams" (or whatever) with the equivalent descriptive richness and semantic rigor for the network.

- o Tools may have to be invented in each organization to map to the culture

5.12.7 Sociological Implications

- Will take a long time to change—mainly due to dollars invested and age of the installed portfolio

- "Let a thousand flowers bloom" needs to end without destroying its benefits

- Technologists must:

  — Take "systems" view
  — Sell its benefits
  — Facilitate cooperation of specialists

5.13 POSTSCRIPT

Although this completed all of the work that the panel set itself to accomplish, reflecting on the 2 days work produced some very interesting, if not profound, overall observations.

In spite of the focus on integration in the workshop, the practical fact of the matter seems to be that: Information Systems are DIS-INTEGRATING—not integrating!!

- Existing systems are decaying through constant maintenance (patching, incremental change). Very likely, in the near future, we will see some major catastrophes!

- The cost of incremental change is accelerating, leaving less of the development dollar available for "building new."

- Users have lost faith that I/S can produce anything useful within an acceptable schedule at an acceptable cost.

- Technology price/performance escalation allows users to "do it themselves," resulting in isolated, (dis-integrated) "islands of automation."

- The diversity and proliferation of technologies defy "connectivity" (integration)!

- Integration is perceived negatively by the business and I/S alike in that it is perceived to be "centralization" and "control," which runs counter to the current trends to decentralize control to the lowest levels of the organization.

- Dis-integration may even have cultural origins in the Western world where individualism is dominant (in contrast to the Eastern world where the orientation is to the group).

- ... and so on !

In contrast, the very concept of "Enterprise" is undergirded by integration. The word "enterprise" etymologically comes from concepts that mean "purpose in action." "Integration" means <u>unity</u> of purpose in action. Furthermore, the word "corporation" derives from words that mean "a body that acts as one."

What appears to be happening in information systems, as well as in the enterprise, is that we have "purpose in action," but it is at the sub-unit level, as opposed to the Enterprise level. This constitutes <u>DIS</u>-integration. The "pieces" are, at a minimum, uncoordinated, and are likely to be in conflict with one another. This results in "the sum of the parts being <u>less</u> than the whole." The question that this raises is, "is it possible for such an enterprise to compete effectively with an enterprise in which the sum of the parts is <u>greater</u> than the whole?" (Probably not!)

As global markets evolve with international competition and an unprecedented increase in the rate of change, it would seem that survival may well be dependent on maximizing asset leverage, minimizing the resource dissipation of sub-optimization, and structuring for flexibility in response to the markets and the competition.

<u>Therefore, it would appear that this is a time of great risk—great risk, not only for I/S, but great risk for the Enterprise as a whole!</u>

The two days work on integration, through the analysis based on the "Framework for Information Systems Architecture," clearly leads to the following conclusion: The parts (by any definition, information systems or the Enterprise) are, in fact, or at least should be, <u>integrated</u>. Otherwise, they would not be included as a part of the whole at all!

Further, it is only through integration that it is possible to achieve advances in productivity, asset leverage, quality, flexibility for assimilating changes, and so on. These are the characteristics that are mandatory for operating in the dynamic, future marketplace so universally forecasted by social and economic prognosticators alike.

In short, it would appear that survival in the future, in which change is the predominant environmental characteristic, will be dependent upon <u>integration</u>—which is apparently <u>in</u>consistent with the current trends!

The singular conclusion is that the challenge to I/S professionals is not merely to design the methodologies and build the tools that make integration feasible and practical (that is, cheap and easy), but to introduce the cultural change—to counter the dominant trend—to establish the precedent of employing integration, not to restrict, but to release—not to immobilize, but to make flexible—not to control, but to <u>SERVE</u> the Enterprise—to enable the parts of I/S and the parts of the Enterprise to function in integration such that the "sum of the parts becomes, in fact, greater than the whole."

Our success in understanding integration, reshaping our own agenda and culture—in <u>SERVING</u> the Enterprise through integration—may well be the dominant factor that determines the destiny of the Enterprise in the dynamic environment of this age.

5.14 REFERENCE FOR CHAPTER 5

[ZAC87] Zachman, J., "A Framework for Information Systems Architecture." <u>IBM Systems Journal</u>, Vol. 26, No. 3, September 1987.

6. THE INTEGRATION OF HETEROGENEOUS COMPUTING ENVIRONMENTS

Sandra Heiler

CHAIRMAN

Biographical Sketch

Sandra Heiler is a member of the technical staff at Xerox Advanced Information Technology (formerly CCA), where she is currently the technical leader of Xerox's project to develop an object model and an object management system to support Engineering Information Systems (EISs). The result will be a system to integrate heterogeneous software tools, DBMSs, and databases. Ms. Heiler's research interests include frameworks for federated systems, object-oriented data models, and database management systems for specialized data types, including engineering, statistical, and bibliographic data.

PARTICIPANTS

John R. Callahan
Ahmed Elmagarmid
Elizabeth N. Fong
Andrew Frank
William Kent

Amrish Kumar
Amit Sheth
Walter Simonson
Jacob Stein

6.1 INTRODUCTION

This chapter presents the report of the workshop Panel on the Integration of Heterogeneous Computing Environments. The panel discussions focused on mechanisms for producing federated systems, that is, systems comprised of autonomous software components and heterogeneous databases. This is in contrast to systems where homogeneity is induced by modifying the components to embed knowledge of the other components in them and standardizing their interfaces.

We believe that federation is the only practical approach to configuring large computing environments that must accommodate a legacy of existing systems and be extensible to incorporate new components. In a federation, components are coupled to allow them to exchange information and provide services for one another. They are made interoperable in order to provide the benefits of integration without requiring that they be modified internally.

The emphasis in this report is on problems of information integration, primarily on semantic issues because they are the least-understood aspects of the integration problem. In general, the report assumes that most physical aspects of integration, including hardware interconnection and low-level communications standards, are solved (or nearly solved) problems.

Section 6.2 describes various aspects of heterogeneity in computing environments and why they present persistent problems. Section 6.3 considers various models of integration that range from loose coupling of autonomous components, sometimes called "interoperable" systems, to tight coupling or "integrated" systems. Section 6.4 presents an architecture for federated systems that comprise heterogeneous components, focusing first on the meta-model and schema, then on integration services that must be provided outside of the components. Section 6.5 presents some conclusions.

6.2 PROBLEM STATEMENT

In integrated computing environments, systems communicate to exchange information or get services from one another. Heterogeneity among software systems and databases causes problems when it results in incompatibilities that prevent systems from communicating. The problems are particularly serious when systems appear to communicate but there are misunderstandings in the interchange, e.g., data are misinterpreted or incorrect services are provided. These problems are related to the inability to communicate the semantics of data or services.

Mechanisms to integrate heterogeneous systems, i.e., to establish effective communications among them, must meet four requirements:

- Accommodate existing systems. Installations have costly investments in existing environments that include not only hardware and software systems and databases but also user training. These systems are unlikely to be replaced all at once, by new, integrated systems. Modifying systems in such environments can cause costly maintenance problems if the systems were developed locally; modifications can be infeasible if the systems are vendor-supplied.

- Preserve the specialized nature of software and databases. Certain applications and databases are inherently heterogeneous. For example, graphics systems deal with specialized hardware; engineering tools manipulate specialized design representations. Modifying them to make them compatible would result in a "lowest-common-denominator" approach that would deprive them of needed features or in an enormously complex approach that addresses every feature from every system.

- Extend to new systems. New components cannot necessarily be constrained to be compatible with existing or standardized systems. To do so would stifle innovation or at least restrict available choices. Such restrictions may eliminate choices that use new hardware or programming language implementations or avoid new concepts because they cannot be accommodated in the schema of the existing environment.

- Avoid depending on complete standardization. Standards take so long to establish that often they are impractical as the only means of integration. Furthermore, competing standards exist and will continue to exist for many domains. In addition, conversion of existing software and databases to meet new standards is often impractical or very costly.

We view heterogeneity among systems and databases as not only inevitable but in some cases actually beneficial. We are concerned with providing mechanisms for hiding the heterogeneity among systems when it interferes with their ability to exchange information or request services from each other. That is, our goal is not to make the systems homogeneous but merely to remove or make transparent incompatibilities that preclude effective communications.

Heterogeneity occurs in both physical and logical aspects of systems and databases. Components may be syntactically incompatible, i.e., have different formats, interfaces or names, and they may be semantically incompatible, have different meanings.

One can classify the differences among systems and databases according to the various aspects of the computing environment that are involved, e.g., hardware platform (instruction set, data representation), communications system (protocols), operating system (file names, transaction management, inter-process communication), data (representation, access methods), applications (naming conventions, algorithms), and model (execution paradigm, schema structures, semantics).

The ability to integrate heterogeneous components is based on the specification of agreements between users of the components and among the components themselves. Existing engineering solutions to integration problems mainly address physical incompatibilities because agreements on physical aspects are easily specified. For example, physical aspects of databases are described using the constructs of a data model. This results in a schema that defines the structure of the data and constraints on using it.

On the other hand, few constructs are available for expressing agreements on logical aspects of components. The "behavior" of a component is expressed through algorithms or procedures; the "meaning" of a component is expressed in natural language or not expressed at all, just understood among users. Semantic data models provide some facilities for expressing semantics, e.g., generalization and aggregation relationships. However, they are not adequate for describing procedures or deeper meanings. Object models attempt to address some of these deficiencies but they, too, are not completely adequate.

The result is that data is difficult to integrate and applications are even more so. In neither case can we really capture the semantics of the information or procedures, except in very limited ways through common naming conventions or agreements on the meanings of built-in relationships. What is needed are formalisms for describing more aspects of components so that agreements can be expressed and interpreted by the integration mechanism.

6.3 COMPONENT AUTONOMY AND COUPLING

The components of a federated system can be coupled in various ways that make certain assumptions about the autonomy of the components. The goal is to preserve autonomy, but to exploit whatever commonalities exist among the components. Tightly coupled components use common protocols, share data, and "cooperate" in performing tasks. Loosely coupled components, on the other hand, may use incompatible protocols and data representations, communicate only indirectly, and execute independently. In general, object-oriented systems, which support encapsulation and data abstraction as ways of dealing with complexity in large systems and of providing modularity and code reuse, are associated with loose coupling.

The level of autonomy of a component affects "ownership" of its procedures and data, how it communicates with other components of the system, and how it encapsulates its semantics. We can characterize component autonomy along three axes:

- o Design autonomy results in systems that have different protocols and data structures.
- o Execution autonomy allows each component to decide whether to execute a given request, when to execute it, and how to execute it.
- o Communication autonomy allows each component to refuse to communicate, i.e., to be inaccessible.

Within a federated system, pairs of components can be coupled in ways that range from very loose, where each component is autonomous with respect to the other, to very tight, where each component interacts directly with the other. Notice that the degree of autonomy need not be uniform across the whole federation. It applies to subsets of the components of the federation and determines how those components communicate with one another.

The mode of coupling components affects the number of translations required to exchange information between them. Zero translations are needed when both components use the same representation. One translation corresponds to point-to-point mapping; data produced by one component is translated directly into the format required by the other. Two translations are needed when systems translate into neutral formats, i.e., one to and one from the neutral format (a star configuration). Even more translations are conceivable, e.g., if there are multiple neutral formats or multiple translations required by communication links.

Loosely-coupled systems are the most modular; they provide the best means of dealing with complexity in large systems. They are also usually far easier to maintain than tightly coupled systems because changes to the implementation of a component are unlikely to affect other components. In addition, they are often the most responsive to user needs since interactions among components are controlled by the user.

However, loosely-coupled systems necessarily involve more user knowledge of the system architecture and the characteristics of the components. In addition, they may result in inconsistencies, since there is no central authority to guarantee correctness. Furthermore, they may have performance problems due to excessive translations.

Tightly-coupled systems behave more like integrated systems. The coupling is more transparent to users, who may, in fact, be unaware of the components. In addition, they are more consistent in their use of resources and in their management of shared data. They may also provide better performance than loosely-coupled systems.

However, they must provide a large suite of utilities to help build, manage and maintain coupled components because components are inter-dependent. Changes to one are likely to affect others. This will be a particular problem for very large systems or systems whose components come from different sources, e.g., different vendors. Furthermore, they require central control mechanisms that are outside of the user's control (and maybe outside of his knowledge).

The various forms of coupling and associated component autonomy affect all aspects of the system architecture and execution, particularly with respect to data translation, sharing system resources, and handling of constraints. In addition, they affect the role of the meta-model whose constructs are used to describe the components.

In a loosely coupled arrangement, the meta-model is primarily descriptive; its purpose is to describe, but not to influence the implementations of the components. It "bottoms-out" in procedures and data structures that are determined by the components, without regard for compatibility. That is, the implementations of the components are external to the type system of the meta-model. They are not part of its implementation nor were they generated by it.

In a tightly coupled arrangement, the meta-model must become more prescriptive; i.e., it determines the procedures and data structures of the components, or at least ensures that they are compatible. We describe the role of the meta-model in more detail in the next section.

6.4 ARCHITECTURES FOR INTEGRATING HETEROGENEOUS COMPONENTS

In general, federated systems have three types of elements:

- o the set of software systems and databases that comprise the underlying components.
- o a description of the components provided by a schema. The schema is defined using the common terminology of a meta-model.
- o a set of integration (or "glue") services. These services provide facilities that cannot be provided by individual underlying components because they span components.

The first are the systems that are integrated in the federation. The second and third comprise the integrating mechanisms. They are described below.

6.4.1 The Role of the Meta-Model and the Schema

Systems to integrate heterogeneous components need to capture agreements among the components on their semantics, i.e., on all aspects of the meaning of data and procedures that the systems incorporate. Usually, this meaning is defined in a context. Exchange across contexts leads to misunderstandings because the contexts themselves are not exchanged. The goal of capturing these semantics in a formal way is to avoid these misunderstandings.

The meaning of information includes, for example, its definition (inclusion/exclusion of special cases), units, quality (precision, as-of, ...), algorithms, and pre- and post-conditions of execution. Capturing the meaning of information requires describing both the structure and behavior of the data and procedures that comprise it, i.e., both information content and results of applicable processes.

The role of a meta-model is to formalize descriptions of information and processes to be shared in the integrated environment. These descriptions are "schemas." The meta-model is a set of concepts and terminology for expressing schemas, e.g., the relational model, entity/relationship model(s), and object/function model(s).

The schema provides a common formal description of the semantics of components. Its purpose is to facilitate mappings from the semantics of one component to another. Correct mappings in turn depend on correct understandings of the formal constructs in the schema. "Semantics" ultimately reside in the interpreters of the formalism.

While it might be desirable in many ways to have a single meta-model, i.e., a single formalism for expressing all schemas, multiple meta-models may need to be supported. One consequence of autonomy is that it simply may not be possible for all participants to agree on a single meta-model. However, if the environment is partitioned into sub-domains, a meta-model may only need to capture the union of capabilities in a sub-domain.

Every pair of sharers/communicators must be covered by some common schema. The extreme case, though quite unlikely, would be that each pair was covered by a different schema, with each using a different meta-model. Multiple meta-models are tolerable so long as mappings between them are available and the number of meta-models is small. This is a point-to-point translation paradigm, in contrast to the star paradigm that would be provided if there were a single "master" meta-model.

The state of the art in meta-models is more advanced for data description than for process or procedure description. Relational, entity-relationship, and semantic data models are in common use to describe data. In the emerging object/function models, the distinction between data and procedures is blurred, with information being described as much in behavioral as in structural terms. However, though object/function models may well emerge as the best candidates for a single universal meta-model, there is little consensus among model developers as yet regarding the formalism for the procedural aspects.

Integration of information and processes in the federation involves establishing mappings between local descriptions (i.e., local schemas) and the schema that covers the sub-domain. Such mappings are probably not automatable, since existing descriptions often do not contain enough semantic information. The mapping process will need to be augmented with additional descriptors and hand-crafting of some mapping procedures.

The meta-model must also provide constructs allowing schemas to express information needed by the integration services described below, such as optimization, transaction management, data/structure conversion, etc.

6.4.2 Integration (Glue) Services

Integration services are facilities that form the "glue" that integrates the components. They cannot be provided by the underlying components because they span components. That is, they support the coupling of components. Such services are not necessarily global; their scope may be restricted to subsets of components.

The role of integration services is to provide mappings between the descriptions of the components (using the meta-models) and to provide services that need knowledge of more than one component, e.g., transaction management. In general, the form of coupling between systems determines how powerful the integration services must be. Tight coupling requires powerful integration services, whereas loose coupling tolerates some inconsistencies and imposes fewer controls over components so the integration services can be more limited.

Other non-local services may provide capabilities to enhance performance of the federated system, e.g., caching or replication services and optimizing transformations on requests that span components. Still other services provide for transparency for end users, for example, global name space management or user interface facilities that give the same "look and feel" to interacting with any component.

The following is a list of integration services that are useful for producing federated systems. The required power of the facility is usually determined by the tightness of the coupling it must support.

- o <u>Request execution, process control, and planning</u>. These services include invoking procedures provided by components and controlling their execution, e.g., determining when they terminate, passing back exceptions, and passing parameters. Planning may include selecting an appropriate host for executing a request, based on data availability or load factor, and synchronizing requests to increase parallelism.

- o <u>Name space management</u>. This service involves binding names to resources, mapping between incompatible name spaces of different components and providing a uniform naming mechanism across components at the level that is visible to the end user.

- o <u>Request decomposition and mapping</u>. These services map operations requested by the end user or a software component into requests against other components of the system. An example is the decomposition of a query into sub-queries and translation into appropriate languages for handling by component database management systems.

- o <u>Translation and data structure conversion</u>. These facilities translate data representations from one application's format to another's. They may have to add, remove, duplicate, or reorder the data transmitted between components. Conversion services are often provided in the form of algebras for expressing transformations on various data representations.

- Schema management. These services must provide for maintenance of the schemas provided by the meta-models (i.e., the component-spanning schemas). In addition they must provide utilities for schema development through schema integration services that handle, for example, name inconsistencies and model differences.

- Transaction management. These services synchronize requests to ensure that intermediate results, which may be inconsistent, do not cause global consistency errors or have unpredictable effects. In addition they support recovery after failures. Transaction management services require knowledge of all applications sharing information. For collaborative applications (e.g., engineering and design systems) they may have to allow controlled sharing of intermediate results and support internal communications among components.

- User interface. These provide services for interacting with end users. The services give a common look and feel to various component interfaces. They must provide various styles of interaction that are compatible with the applications, yet appear somewhat consistent. Moreover, they must adapt to various hardware devices.

6.5 CONCLUSIONS

Integration is based on agreements between participants in a federation. Physical integration is a solved or nearly solved problem because formal methods of specifying agreements at this level exist, e.g., communications standards. Logical integration is much harder because adequate formalisms for expressing semantics are not available. Describing semantics (or "meanings") requires describing behavior, which is usually embedded in procedures.

Meta-models provide description facilities for components to support mappings between them. Existing models suffice for describing information structure. However, they are weak in behavioral specification capabilities. Object/function models are the best candidates to provide solutions in the future.

Autonomous systems are independent of other systems in a computing environment. They provide needed modularity for large, complex computing environments. Federations of autonomous systems are both easier to develop and easier to maintain and extend than systems where components are interdependent.

The mode of coupling systems determines their degree of autonomy. Loose coupling allows components to remain autonomous. However, loosely coupled systems lack the controls of tightly coupled systems. They are prone to inconsistencies and may perform worse than tightly coupled systems. The optimal degree of integration is probably not at the extremes of loose or tight coupling, but someplace in the middle. Moreover, most systems will mix styles of coupling among components.

Integrating services becomes harder to provide as systems are more tightly coupled because they must provide more control and be based on more knowledge of the internals of the underlying components. They must support direct communications between components and ensure consistency among their results. Integrating services for loosely coupled components can provide generic facilities that are customized based on formally specified external descriptions of the components. Integrating services for tightly coupled systems must be based on knowledge of the internals of specific components.

7. ARCHITECTURES AND STANDARDS

W. Bradford Rigdon

CHAIRMAN

Biographical Sketch

Brad Rigdon is the Senior Vice President, Research and Development, for McDonnell Douglas Information Systems Company (MDISC).

Mr. Rigdon entered the data processing field in 1961, joining McDonnell Automation Company (McAuto) a year after it became a division. By 1966, he was a Senior Consultant. He then held a variety of management positions in systems services and marketing in New York, including Regional Director of the Northeast Region. He has also been a Director of Product Planning, Director of Product Management, and Director of Business Applications for McAuto in Long Beach. In 1980 he was named Director of McDonnell Douglas Corporation Services, in 1983 Vice President of the MDC Services organization of McAuto, in 1986 General Manager of McDonnell Douglas Aerospace Information Services Company (MDAIS), and Senior Vice President—Special Projects for MDISC in 1988, and Senior Vice President—R&D in 1989.

Mr. Rigdon was graduated in 1961 from the University of Missouri with a bachelor's degree in business administration. He is currently a member of the Executive Committee of the Regional Consortium for Education and Technology and President and Chairman of the Board of Product Definition Exchange Standard (PDES, Inc.).

PARTICIPANTS

Robert J. Benson
Helen T. Cohn
John W. Coyle
John W. Eichler
Kenneth Gordon
Asaph H. Hall
Merrill M. Hessel
Ray Hollenbach
Jack H. Jones
Marilyn Parker
Bruce Rosen
Robert Smith
Charles Tuller
Jim Wick
Jerry Winkler

7.1 OVERVIEW

This panel addressed the role of architectures and standards in supporting information management throughout an enterprise.

This paper addresses the following issues:

- Levels of architecture in an enterprise
- Problems addressed by architecture
- Benefits and risks of having architecture.

The paper also includes a table (figure 7-2) with specific examples of the standards within different levels of architecture.

Architecture discussions frequently focus on technology issues. This paper takes a broader view, and describes the need for an "enterprise architecture" that includes an emphasis on business and information requirements. These higher level issues impact data and technology architectures and decisions.

Standards are used to implement and enforce the architecture. There are mandated standards that are required by regulation or the customer (e.g., IRS, OSHA); standards that have been agreed to by industry or national and international bodies (e.g., ISO, ANSI); and standards developed and promulgated within the individual enterprise (e.g., PROFS as the electronic mail system).

There is not a single correct way to develop an architecture or implement standards for every enterprise; they must be customized to the environment. For example, military companies must consider CALS and the various MIL specs in developing their architecture and the supporting standards.

7.2 LEVELS OF ARCHITECTURE

Architecture is defined as a clear representation of a conceptual framework of components and their relationships at a point in time.

In this definition, a component is an element or item addressed by a particular architecture, and relationships are the connections between the components.

Any discussion of architecture must also consider the role of standards. Standards enable or constrain the architecture and serve as its foundation. They are an essential part of an architecture implementation, but are generally not selected until the architecture has been developed.

The definition of architecture is general enough to encompass planning needs throughout an enterprise. Thus, a discussion of architecture must take into account different levels of architecture. These levels can be illustrated by a pyramid, with the business unit at the top and the delivery system at the base (figure 7-1). An enterprise is composed of one or more Business Units that are responsible for a specific business area. The five levels of architecture are:

- Business Unit
- Information
- Information System
- Data
- Delivery System

The levels are separate yet interrelated. The first four are related in a top-down dependency, as discussed below. The fifth level, the Delivery System architecture, is the foundation architecture: it is created to meet the requirements of the other architectures. A successful Delivery System architecture is dependent upon the definition of relevant business goals and objectives.

The idea of an enterprise architecture reflects an awareness that the levels are logically connected and that a depiction at one level assumes or dictates that architectures at the higher levels have been completed.

An architecture is a description of one of these levels at a particular point in time. It may represent a view of a current situation with islands of automation, redundant processes and data inconsistencies. It can also be a representation of a future integrated automation information

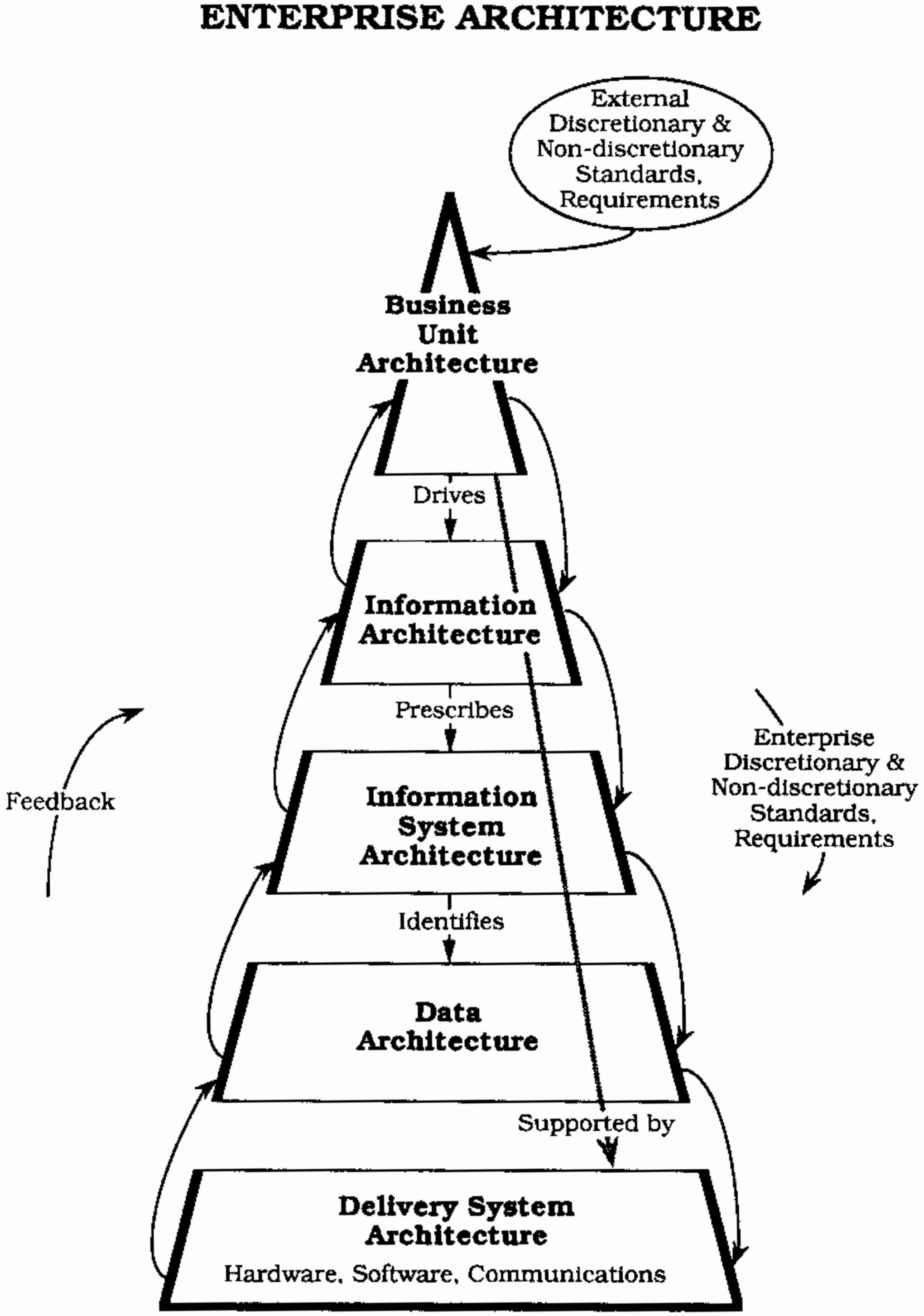

ENTERPRISE ARCHITECTURE
External Discretionary & Non-discretionary Standards, Requirements
Business Unit Architecture
Drives
Information Architecture
Prescribes
Information System Architecture
Identifies
Data Architecture
Supported by
Delivery System Architecture
Hardware, Software, Communications
Feedback
Enterprise Discretionary & Non-discretionary Standards, Requirements

Figure 7-1

structure towards which the enterprise will move in a prescribed number of years. An architecture of the current (or "as is") state is an important step in the development of an "end state" architecture that gives context and guidance for future activities.

The following descriptions of each level of architecture apply to most large enterprises, with the recognition that any level may be uniquely enabled or constrained by industry or government standards, as well as by internal policies and procedures.

The Business Unit may portray either a total corporate entity (that is, the enterprise is the business unit) or a corporate sub-unit. Architecture at this level establishes a framework for satisfying both the internal information needs and the information and data needs imposed by external organizations. These external organizations include cooperating organizations, customers, and federal agencies. The information and data needs at this level impose requirements to be satisfied by lower levels of the architecture, with increasing attention to technical considerations.

The representation of the Business Unit architecture shows organizational units and their relationships and business processes and their relationships, as well as specific standards, policies, and procedures that enable or constrain the accomplishment of the overall enterprise mission.

The Information architecture establishes a framework to meet the information needs of the Business Unit architecture. The Information architecture specifies the content, presentation form, and format of the information, thus establishing requirements for the Information System architecture.

The representation of the Information architecture should relate the information sources and uses with the organizations that generate or use either internal or external documents, data, etc. This level should represent technical and management information flow, as well as the impact of time on information integrity and meaning.

The Information System architecture establishes a framework for meeting the specific information requirements given by the Information architecture. This architecture uses its components to acquire and process data, then to produce and distribute information in accordance with the Information architecture requirements and standards.

The representation of the Information System architecture shows the automated and procedure-oriented information systems that support the internal and external information flow. Logical database designs occur at this level.

The Data architecture establishes a framework for maintenance, access, and use of the data of the enterprise. The data should meet the standards of the Business Unit and the other upper levels of the architecture. The creation of a data dictionary and associated naming conventions is an important aspect of the Data architecture, because these conventions establish the vocabulary necessary for communication among the human elements of the business unit.

The representation of the Data architecture relates the data that supports the information systems structure. This will include data models that support physical database design; database and file structures; and data definitions, dictionaries and data elements that underlie the information systems of the enterprise.

The Delivery System architecture is a technical implementation to meet the requirements of all higher levels.

The representation of the Delivery System architecture describes the computer and communication hardware and software required to support the data and information systems levels of the enterprise architecture. It also describes the infrastructure and facility support requirements to properly accommodate and connect these assets in an integrated manner.

Figure 7-2, Sample Elements of an Enterprise Architecture, gives examples of the following categories for each level of architecture:

- Components
- Nondiscretionary standards to which an enterprise must adhere
- Discretionary standards that an enterprise may select as part of its architecture

Sample Elements of an Enterprise Architecture

Level	Component	Nondiscretionary Standards	Discretionary Standards
1. Industry	--	--	Data Interchange (PDES, IGES, EDI)
2. Business Unit	- External reporting requirements - Functional areas - Internal corporate reporting requirements	IRS Business practices Legislation Military procurement	Policies: - Use industry, Int'l, Nat'l, DoD, company precedence - Use standards that provide: • reuseability of assets • migration from current environment • evolution in the future • information sharing
3. Information	- Composite documents - Data - Transformations - Responsible organization	Gov/industry reporting formats and forms	- Security - Integrity (rules of use) - Naming conventions, semantics - Description methods - IRDS - SGML
4. Info systems	- Specifications - Requirements - Applications, "modules" - Databases - Procedures	DoD software development standards	- CASE, object-oriented approaches - SQL, NDL, E/R - Languages - CASE Repository - IRDS - Word processing, spreadsheets - Methodologies - Graphics (GKS, CGM)
5. Data	- Elements - Structures - Data model		- Naming conventions - Representation - File, data base compatibility (interprocess sharing) - IRDS
6. Delivery Systems	- Communications - Operating Systems - Hardware - Utilities and tools - Data storage media		- OSI - GOSIP - Interchange stds (PDES, IGES, ASN.1) - Hardware architectures - Optical storage

Figure 7-2

Note that the table begins with an additional level, Industry. This level is included as a recognition that certain standards exist above those in a specific enterprise or business unit.

7.3 PROBLEMS ADDRESSED BY ARCHITECTURE

Creating an architecture for each level of an enterprise enhances the enterprise's ability to guide decision-making, to manage change, and to communicate the organization's business goals, objectives and policies up and down its hierarchy and across its functional components.

7.3.1 Guiding Decisions

Architecture provides the framework for ensuring that enterprise-wide goals, objectives, and policies are accurately reflected in decision-making related to building, acquiring, or changing information systems. Without appropriate architectures, there is no assurance that standards for interprocess communication, data naming, data representation, data structures, and information systems will be consistently or appropriately applied across the enterprise. In addition, the organization will not have:

- An enterprise-wide conceptual framework for planning information systems development

- Appropriate principles, criteria, and standards that are agreed to by the business and can be used to measure existing applications against proposed upgrades or changes

- A relatively stable framework of agreed principles, boundaries, and objectives within which specific information systems can be developed and implemented cost-effectively

- Standards and guidelines to measure conformance of commercially available packages and tools in make-or-buy decisions

- Agreement within the organization about which applications, data, and interfaces are the targets for implementation within a specified time-frame

The lack of architectures produces a conceptual vacuum when decisions must be made about organization funding and priorities for the Information System. Also, opportunities for common reference files and applications, corporate directories, common communications capabilities, data naming, data structures and definitions, and other infrastructures cannot be optimized for the enterprise as a whole without architectures.

7.3.2 Managing Change

Managing change has become a pervasive issue, particularly as the penetration of information technology quickens the pace of change within enterprises. A key issue in dealing with the high rate of change in today's complex information environment is understanding the effect of any given change in the context of the enterprise as a whole, rather than in terms of its components.

The context and perspective for evaluating simultaneous and often interrelated changes is provided by an architecture that consists of the components and their relationships, and the enterprise's goals, objectives, policies, and standards. In the absence of an architecture, an enterprise cannot effectively deal with change and may not be able to understand the impact of change within the enterprise and across its components.

The first principle for managing change is an accurate representation of the status of the enterprise's goals, objectives, policies, and standards as they apply to information systems prior to and after change. The architecture must be capable of accurately reflecting the status of information systems across the enterprise, and the interrelationships among them, at different points in time.

7.3.3 Improving Communications

With a well-defined architecture, the enterprise's business goals and objectives will be known consistently across the functional components of the organization. Lack of an architecture leads to information systems with no clear picture of the interrelationships among the systems. Organizational components will develop systems independently and not adequately communicate about the linkages between their information systems. The consequence is missed opportunities to share data and, worse, it can result in the different definition and different naming of the same data, or the same naming of different data.

The lack of communication about goals and objectives can produce duplicate information systems. Not only are communications across functions impaired, but communications up and down the organization will not be adequate to ensure that information systems are developed in conformance with the essential business goals and objectives of the enterprise. The consequence can be information system projects that are seen as not meeting requirements, redirection of projects, slipped schedules, overrun development costs, reworked systems, and general dissatisfaction with information systems progress.

In a worst-case scenario, lack of an architecture can result in investments in "hobby horse" information systems that do not satisfy the goals and objectives of top management but instead reflect only the more narrow goals and objectives of a subcomponent of the enterprise.

7.4 BENEFITS AND RISKS

Underlying the following statements about benefits and risks is a basic principle: architectures—at whatever levels—are important only to the extent they link to and enable success in the basic mission and performance of the organization.

Consequently, the thrust of the benefits and risks described here is to measure and define them according to their relationship to the underlying business being served. The benefits and risks are classified into three categories:

- consistency with business planning
- communication within the organization
- cost or economic impact.

7.4.1 Consistency with Business Planning

The enterprise has objectives (e.g., be low-cost provider, increase market share, create a new market) and a strategic plan to meet these objectives. To the extent that the architectures support the business strategies of the enterprise and enable rapid change, benefits will accrue. When the architectures constrain the business activities, risk is increased.

Benefits include:

- o Linkage of information system investments to business purposes

  The process of doing architecture forces a recognition of business purpose, and consequently a linkage of individual projects to recognized business purpose. Architectures give a clearer sense of priorities among information system investments linked directly to changes in business processes caused by changing business conditions. By linking the information systems investments to the business purpose, as conditions change the enterprise can maximize the usefulness of these investments by synchronizing implementation with the business change.

- o Ability to follow-on from proven architectures

  An enterprise may choose the strategy of following the lead of others in their industry by implementing proven architectures at a lower cost point in the maturity cycle, and thus maintain a cost comparability with its competition. In this case, the enterprise sacrifices early achievement of benefits and potential competitive advantage for reduced investment cost and maintenance of market share.

In certain business environments, market dynamics demand a high degree of organizational flexibility to service the rapidly changing market. To the degree that the architectures support flexibility and change, benefits accrue.

Risks related to consistency with business planning include:

- o Responsive reaction to changes in business plan

  Business changes may occur so rapidly that the architecture cannot adequately anticipate the change with new information systems investments.

- o Short-term solutions with long-term support cost

Short-term solutions in a specific area may dictate purchase of a specific vendor application package. The architecture limits flexibility in this area because data and other standards may be incompatible with the package. Thus short-term business expediency can create a long-term information system support requirement.

7.4.2 Communication Within the Organization

Architectural policy decisions at the enterprise or business unit level will determine how major business processes will be conducted. This implies cross-functional decisions rather than unilateral, disconnected decisions by individuals or separate functional units. This can cause significant process improvements such as:

- Reduced span time
- Elimination of redundant activities
- Increased quality
- Reduced product cost

Benefits include:

- Resolve conflict

  The process of establishing an architecture (at any level) and of selecting the standards that support or implement that architecture is based on achieving consensus on architectural issues. Successful results require agreement on key business requirements, on strategies to satisfy those requirements, and on how to implement those strategies.

  Even if the architecture is not completed, the understanding and agreement resulting from the process resolves conflict between organizational entities. Implementation puts the agreements in place.

- Flexibility for future business changes

  An architecture that is designed to allow for future changes in products and business processes will enable an enterprise to respond to market shifts and technology breakthroughs more quickly and at lower cost. This implies the need for open technical architectures and use of industry standards rather than proprietary standards.

- Compatibility

  Defining the architecture to allow for compatibility with suppliers, customers and team members will provide competitive advantage for the total group, versus a situation where the units are neither connected nor compatible.

- Communicate information

  If the organization can better understand information bottlenecks through the analysis of the integrated levels of the enterprise architectures, much can be done to resolve and eliminate those bottlenecks.

- Identify what standardization is required

  Similar methods or processes should be implemented in similar ways. An architecture assists in identifying candidates for standardization.

- Facilitate cross-functional decisions

  Architectures result in better definition of interrelationships among information systems, reduced incompatibility of data, and the identification of dysfunctional structures in the organization.

- Communications

  A major purpose of the architectures is to facilitate rational communications among organizations with different perspectives on how they fit and contribute to the successful accomplishment of organizational goals.

  Communications are frequently aimed at solving current critical problems. Within a structural framework based on mutually understood goals, communication can be used to avoid such problems through integrated planning and team-directed action tactics.

o Define relationships

Architectures can transform the way an enterprise functions. Such transformations generally cross organizational boundaries and produce environments where the boundaries themselves may change. The relationships needed to proactively enable such change are defined, and possibly coordinated, using the architecture as a vehicle.

The business processes and information intersections identified in the architectural process become the basis for defining the working relationship, responsibilities, and dependencies within the organization.

Information-oriented relationships that exist across departmental boundaries are defined in the process of developing a Data architecture. As a consequence, data sharing and data ownership is understood and properly designated. Consequences of actions or changes can be more predictably anticipated.

o Evaluation criteria for applications

The architecture and its associated standards provide a means of evaluating applications and designs for conformance to the standards. The more detailed the architecture, the more specific and non-subjective can be the evaluation.

o Insulate changes

To the extent that an architecture defines interfaces between modules, the underlying code, data structure and hardware can be replaced while retaining the interface standard. Thus, upgrades can be continuous and isolated from each other, rather than requiring an all-or-nothing change.

Risks related to communications within the organization include:

o Acceptance of cultural impact

Often absent from the technology planning and implementation process is consideration of the human impact and organizational dynamics. The risk in overlooking this aspect is that new tools are provided to an unreceptive (perhaps hostile) audience. When this happens, the benefits of the new process or technology may never be realized.

- Control by MIS or user management

  Architecture and standards could be used by the MIS organization to unnecessarily limit choice by the user. It could also be used by short-sighted management to limit the cost of implementation and operation at the expense of organizational effectiveness and flexibility.

- Premature standards selection without accommodating change

  The business requirements can dictate early adoption of internal standards that affect formats, data, objects, or remote procedure calls that are implemented prior to international or industry standards being established. The more these are incorporated into designs and the more they differ from the ultimate standards, the more difficult the subsequent transition.

- Premature implementation

  The early implementation of untested technology and standards frequently causes problems with raised expectations that will not be realized.

7.4.3 Cost or Economic Impact

Economic impact is a function of cost reduction or revenue enhancement for the business organization. Such quantifiable benefits and risks related to architectures are expressed in the accepted investment justification approach used by the enterprise, e.g., ROI, discounted cash flow, etc.

Benefits include:

- Reusability

  A reusable component of the architecture is potentially a standard component, thereby reducing or eliminating redundant effort. Reducing redundant effort can reduce cost to the organization.

Reusable components that provide the same or similar functionality can best be identified through an architecture. These reusable components should be developed and structured to support all the overlapping functions and not be totally separate components.

- Reduced development and maintenance costs

  Development and maintenance costs can be reduced where a pre-defined architecture provides guidance and a framework. For example, architectures reduce duplication of equipment (hardware) and functions through clear architecture design; reduce maintenance through selection of standards and appropriate tools; reduce software costs by using neutral data exchange formats thus eliminating individual translators; and improved quality through conformance to standards and architecture.

Economic impact can be negative as well, where potential risks or additional costs are not recognized. These risks include:

- Premature adoption

  Premature adoption of standards frequently leads to unnecessary investments and the need to reinvest in later technologies.

- Initial investment

  There will be up-front costs to design, implement, and convert to the architecture and specific standards.

8. PARTICIPANTS

The following is the list of Information Management Directions 5 participants:

Daniel Appleton
General Chairman
D. Appleton Company, Inc.
1334 Park View Avenue, Suite 220
Manhattan Beach, CA 90266

Robert J. Benson
Architectures & Stnds.
Washington University
Campus Box 1220
St. Louis, MO 63130

T. N. Bernstein
T & B Data Mgmt.
U.S. Air Force
Wright Aeronautical Laboratories
AFWAL/FIBLL
Wright Patterson AFB, OH 45433

Jean Berube
Systems PD&M Tools
Assyst Technologies
1080 Beaver Hill - Suite 1400
Montreal, Quebec,
H2Z 158 CANADA

Susan Blair
Knowledge & Data Mgmt.
DataDog
P.O. Box 15512
Alexandria, VA 22309

Olin Bray
T & B Data Mgmt.
Combustion Engineering
51 North Main
West Hartford, CT 06107

Bruce Buckelew
Systems PD&M Tools
IBM Corporation
425 Market Street
20th Floor
San Francisco, CA 94105

Donald C. Burnstine
Systems PD&M Tools
Burnstine Associates
25 Dana Street
Cambridge, MA 02138

John R. Callahan
Heter. Computing Envir.
University of Maryland
Department of Computer Science
College Park, MD 20742

Helen T. Cohn
Architectures & Stnds.
McDonnell Douglas Aerospace
Information Services Co.
Advanced Systems Engineering
W401-304-1E/197
325 McDonnell Blvd.
Hazelwood, MO 63042

John W. Coyle
Architectures & Stnds.
U.S. Department of the Interior
Office of Information Resource Mgt.
M/S 5312, MIB
18th & C Streets, NW
Washington, DC 20240

Robert Curtice
Knowledge & Data Mgmt.
Arthur D. Little, Inc.
35 Acorn Park
Cambridge, MA 02140

Jean-Pierre DeJean
T & B Data Mgmt.
General Electric
55 Merchant Street
Cincinnati, OH 45246

Tom DeMarco
Keynote Speaker
Atlantic Systems Guild
P.O. Box 553
Rockport, ME 04856

John W. Eichler
Architectures & Stnds.
General Motors Technical Center
Advanced Mfg. Engineering
30300 Mound Road
Warren, MI 48090-9040

Ahmed Elmagarmid
Heter. Computing Envir.
Purdue University
Department of Computer Science
West Lafayette, IN 47907

Mary Ann Englebert
T & B Data Mgmt.

U.S. Navy
Data Automation Command
Building 166, Washington Navy Yard
Washington, DC 20374

Douglas Erickson
Systems PD&M Tools

Pacific Power and Light
920 SW Sixth Avenue
Portland, OR 97204

Elizabeth N. Fong
Heter. Computing Envir.

National Institute of Standards
and Technology
Technology Bldg., A266
Gaithersburg, MD 20899

Andrew Frank
Heter. Computing Envir.

University of Maine
Dept. of Surveying Engineering
120 Boardman Hall
Orono, ME 04469

Lawrence E. Frank
T & B Data Mgmt.

Boeing Computer Services
1801 Alexander Bell Drive
Reston, VA 22091

Robert E. Fulton
T & B Data Mgmt.

Georgia Institute of Technology
School of Mechanical Engineering
Atlanta, GA 30332

Alan H. Goldfine
NIST

National Institute of Standards
and Technology
Technology Bldg., A266
Gaithersburg, MD 20899

Dale Goodhue
Systems PD&M Tools

University of Minnesota - Twin Cities
Curtis L. Carlson School of Mgmt.
Management Science Dept.
271 19th Avenue, S.
Minneapolis, MN 55455

Kenneth Gordon
Architectures & Stnds.

Digital Equipment Co.
555 Virginia Road
UR03-2-B7
Concord, MA 01742-2727

Asaph H. Hall
Architectures & Stnds.

General Dynamics Corporation
Data Systems Division
12101 Woodcrest Executive Drive
St. Louis, MO 63141

Sandra Heiler
Heter. Computing Envir.

Xerox
Advanced Information Technology
4 Cambridge Center
Cambridge, MA 02142

Merrill M. Hessel
Architectures & Stnds.

National Institute of Standards
and Technology
Administration A1126
Gaithersburg, MD 20899

Ray Hollenbach
Architectures & Stnds.

General Dynamics Corp.
Data Systems Division
12101 Woodcrest Executive Drive
St. Louis, MO 63141

James Hosmer
Systems PD&M Tools

Lockheed Space Operations Co.
1100 Lockheed Way - M/S LSO-151
Titusville, FL 32780

William Inmon
Systems PD&M Tools

American Management Systems
66 South Van Gordon
Lakewood, CO 80228

Larry L. Johnson
T & B Data Mgmt.

Texas Instruments
6500 Chase Oaks Blvd.
P.O. Box 868305 - M/S 8406
Plano, TX 75086

Randall Johnson
T & B Data Mgmt.

Boeing Computer Services
Boeing Advanced Technology Center
P.O. Box 24346, M/S 7L-64
Seattle, WA 98124-0346

Jack H. Jones
Architectures & Stnds.

McDonnell Douglas Aerospace
Information Services Co.
Advanced Systems Engineering
W400-304-1E-197
325 McDonnell Blvd.
Hazelwood, MO 63042

Beverly Kahn
Systems PD&M Tools

Suffolk University
School of Management
8 Ashburton Place
Boston, MA 02181

William Kent
Heter. Computing Envir.

Hewlett Packard Labs
1501 Page Mill Road
P.O. Box 10490
Palo Alto, CA 94303-0969

Amrish Kumar
Heter. Computing Envir.

Honeywell
Computer Sciences Center MN63-C080
1000 Boone Avenue North
Golden Valley, MN 55427

Don Libes
Knowledge & Data Mgmt.

National Institute of Standards
and Technology
AMRF
Met. Building A127
Gaithersburg, MD 20899

Mary Loomis
T & B Data Mgmt.

Hewlett Packard
19447 Pruneridge Avenue
M/S 44UW
Cupertino, CA 94014-9974

Richard J. Mayer
Knowledge & Data Mgmt.
Texas A&M University
Knowledge Based Systems Lab
Industrial Engineering Dept.
College Station, TX 77843-3131

Anne Miller
Systems PD&M Tools
IDS Financial Services
IDS Tower 10, R8/527
Minneapolis, MN 55440

Linda Nadeau
Systems PD&M Tools
KnowledgeWare
2006 Hogback Road
Ann Arbor, MS 48105

Prasad Nair
Knowledge & Data Mgmt.
Project Management, Inc.
6317 Poe Road
Bethesda, MD 20817

Shamkant B. Navathe
T & B Data Mgmt.
University of Florida
Database Sys. Research & Dev. Ctr.
CSE Bldg., Room 470
Gainesville, FL 32611

Gerry O'Beirne
Systems PD&M Tools
Fireman's Fund
14 Willow Avenue
Larkspur, CA 94939

T. William Olle
Systems PD&M Tools
2 Ashby Park Road
Walton on Thames
Surrey KT12 1JU,
ENGLAND

Gerard Otten
Systems PD&M Tools
DCE Info. Mgmt Consulting
Weverstede 1
B431J5 Nieuwegein,
NETHERLANDS

Marlene A. Palmer
Knowledge & Data Mgmt.
Farm Credit Administration
1501 Farm Credit Drive
McLean, VA 22102-5090

Marilyn Parker
Architectures & Stnds.
IBM/Academic Information Systems
Los Angeles Scientific Center
11601 Wilshire Blvd.
Los Angeles, CA 90025-1738

Judith Quillard
Systems PD&M Tools
Massachusetts Institute of Tech.
Center for Information Systems Res.
77 Massachusetts Ave., Rm. E40-191
Cambridge, MA 02139

Tim Rau
Knowledge & Data Mgmt.
NASA
Space Station M/C 551 Freedom Proj.
10701 Parkridge Blvd.
Reston, VA 22091

W. Bradford Rigdon
Architectures & Stnds.
McDonnell Douglas
Aerospace Information Services
P.O. Box 516
St. Louis, MO 63166

Stephen Ritzman
T & B Data Mgmt.
Booz, Allen Hamilton
11410 Isaac Newton Square
Reston, VA 22090

Bruce Rosen
Architectures & Stnds.
National Institute of Standards
and Technology
Technology Bldg., A266
Gaithersburg, MD 20899

Robert S. Rosenzweig
Knowledge & Data Mgmt.
Fireman's Fund Insurance Company
1600 Los Gamos Road
San Rafael, CA 94911-5800

Gary Schuldt
Systems PD&M Tools
Syncretics, Inc.
7313 1/2 Woodlawn Avenue, NE
Seattle, WA 98115-5334

Amit Sheth
Heter. Computing Envir.
Unisys Corporation
2400 Colorado Avenue
Santa Monica, CA 90406

Charles Shoecraft
Systems PD&M Tools
2460 W. 255th Street
Lomita, CA 90717

Walter Simonson
Heter. Computing Envir.
Defense Logistics Agency
Code ZWF 3A636
Cameron Station
Alexandria, VA 22304

Robert Smith
Architectures & Stnds.
U.S. Department of Transportation
Transportation Systems Center
Cambridge, MA 02142

Richard Smith
Systems PD&M Tools
Arizona State University
Industrial Engineering
2116 E. Geneva
Tempe, AZ 85282

Douglas Snyder
Systems PD&M Tools
IBM Canada
100 Consilium PI.
Scarborough, Ontario,
M1H 3E8 CANADA

Jacob Stein
Heter. Computing Envir.
Servio Logic Corp.
15025 Southwest Koll Parkway
Suite 1A
Beaverton, OR 97006

Stanley Y. W. Su
Knowledge & Data Mgmt.
University of Florida
Database Systems Research
& Development Center
E470 Computer Sciences & Eng.
Gainesville, FL 32611

Charles Tuller
Architectures & Stnds.
IBM Corporation
630/221-2
Boardman Road
Kenyon House
Poughkeepsie, NY 12601

Alvin Underbrink
Knowledge & Data Mgmt.
Texas A&M University
Knowledge Based Systems Laboratory
Industrial Engineering Department
College Station, TX 77843-3131

T. Travers Waltrip
Systems PD&M Tools
The Travelers Corporation
One Tower Square
Hartford, CT 06183

Miles Welter
Systems PD&M Tools
IBM Santa Teresa Lab
555 Bailey Avenue
San Jose, CA 95150

Jim Wick
Architectures & Stnds.
Lockheed Corporation
Information & Admin. Services
4500 Park Granada Blvd.
Calabasas, CA 91399-0350

John F. Willis
T & B Data Mgmt.
Rockwell International
North American Aircraft Division
201 North Douglas Street
P.O. Box 92098 - M/C GC-10
Los Angeles, CA 90009

Jerry Winkler
Architectures & Stnds.
Computer Technology Assoc.
7927 Jones Branch Drive
Suite 600W
McLean, VA 22102

John A. Zachman
Systems PD&M Tools
IBM
355 South Grand Avenue
Los Angeles, CA 90071

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET** *(See instructions)*	1. PUBLICATION OR REPORT NO. NIST/SP-500/167	2. Performing Organ. Report No.	3. Publication Date September 1989

4. TITLE AND SUBTITLE

Information Management Directions:
The Integration Challenge

5. AUTHOR(S)

Elizabeth N. Fong and Alan H. Goldfine, Editors

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
(formerly NATIONAL BUREAU OF STANDARDS)
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Same as item #6

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 89-600755

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

This report constitutes the results of a 3 day workshop on the integration challenge of information management, held in Fort Lauderdale, Florida on October 31 to November 2, 1988. The workshop was sponsored by the National Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), formerly the National Bureau of Standards (NBS), in cooperation with the Association for Computing Machinery (ACM), the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE), and the Federal Data Management Users Group.

This workshop was the fifth in the Information Management Directions series (formerly called Data Base Directions). The purpose of these workshops is to examine, in depth, key trends and strategies that affect the future of the information management profession. The focus of this fifth workshop was on issues related to integration and productivity.

The 72 workshop participants were organized into five working panels, which met to discuss the integration of knowledge and data management; the integration of technical and business data management; the integration of systems planning, development, and maintenance tools and methods; the integration of distributed, heterogeneous computing environments; and architectures and standards for information management.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

architecture; database; database management; DBMS; distributed, information management; integration; knowledge base; standards.

13. AVAILABILITY

☒ Unlimited

□ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

□ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

174

15. Price

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300