EVOLUTION TOWARDS STRATEGIC APPLICATIONS OF DATABASES THROUGH COMPOSITE INFORMATION SYSTEMS

Stuart E. Madnick
Y. Richard Wang

May 1988

CISR WP No. 170
Sloan WP No. 1862-87

Forthcoming in *Journal of MIS.*

Center for Information Systems Research
Sloan School of Management
Massachusetts Institute of Technology

Evolution Towards Strategic Applications of Databases Through Composite Information Systems

Stuart E. Madnick
Associate Professor of Management Science
Center for Information Systems Research
Sloan School of Mangement
Massachusetts Institute of Technology
Cambridge, MA 02139 (617) 253-6671

Y. Richard Wang*
Assistant Professor
Department of Management Information Systems
College of Business and Public Administration
University of Arizona, Tucson, AZ 85721 (602) 621-2748
*Currently on leave to MIT as a Visiting Assistant Professor

ABSTRACT Many important strategic applications involve inter-corporate linkage or intra-corporate integration of information. This requires multiple databases to work together. We refer to this category of information systems as *Composite Information Systems* (CIS). Migrating from separate systems to a more fully integrated CIS environment is usually a difficult, expensive, and time-consuming process both due to technical and organizational realities.

An evolutionary approach is presented in this paper to meet the challenge. The essence of this approach is captured in four CIS principles: (1) the separation of data from processing; (2) the use of flexible tools; (3) the use of interfaces that facilitate data conversion and communication between processing components and databases; and (4) the explicit recognition of the CIS environment. Based on the principles, we delineate five stages of evolution, which may co-exist: (1) separate systems; (2) virtual-driver; (3) logical separation; (4) physical separation; and (5) specialized functional engine.

The opportunities for strategic uses of information technology in organizations are often blocked by the difficulties of getting from the current state to the desired situation. The evolutionary process presented in this paper has been found to be effective in overcoming this problem.

KEY WORDS distributed databases, organizational information systems, strategic computing, systems development.

ACKNOWLEDGEMENTS Work reported herein has been supported, in part, by the Department of Transportation's Transportation System Center, the U.S. Air Force, the U.S. Space and Naval Warfare System Command, the Center for Management of Information at the University of Arizona, and Citibank. The authors also wish to thank the anonymous referees for their valuable comments on an earlier version of this paper.

1. Introduction

Significant advances in the price, speed-performance, capacity, and capabilities of new database technology have created a wide range of opportunities for business applications to meet corporate strategic goals. This paper presents *Composite Information Systems (*CIS*)* as an evolutionary process towards the strategic applications of databases. It provides a framework for the evolution of separate systems to a more fully integrated CIS with value being added to the organization at each stage.

Research background and concepts of CIS are presented in the remainder of this section. Section 2 discusses strategic CIS opportunities. Section 3 examines principles that provide guidelines to the evolution of CIS. In section 4, an evolutionary approach is delineated as an implementation strategy for CIS. Finally, concluding remarks are made in section 5.

1.1 Research Background

Development and deployment of *information technology* (IT) for strategic advantage have become very topical. Rockart and Scott Morton [36] indicated that information availability and communication processes are having a significant impact on corporate life. Using a conceptual model of technology impact in which all the elements of corporate functioning (technology, strategy, organization structure and culture, managerial processes, and individuals and their roles) must be kept in balance, they argued that technology, both externally and internally, is driving the others. Rockart's well-known *Critical Success Factors* (CSF) [35] perspective has provided *information systems* (IS) executives a structured approach to define their own data needs in order to identify potential strategic opportunities. Porter and Millar [32,33] also found that IT is changing the rules of competition for U.S.

industry by: (1) changing industry structure and boundaries; (2) dramatically reducing costs, thereby, creating competitive advantage; and (3) creating new products and services, sometimes spawning completely new business.

Several other approaches have also been proposed to link IT with strategic management. McFarlan and McKenney [30] used a two-by-two strategic grid to show how IT changes the way corporations compete. Wiseman and MacMillan [40] proposed an option-generator with five level of questions to identify strategic IS opportunities in the process of developing a competitive strategy. Beath and Ives [7] combined the familiar Anthony framework and information attributes to produce a methodology for competitive information systems in support of product pricing.

These approaches reflect the wide recognition of IT's potential for strategic computing. It is also increasingly evident that the identification of strategic applications alone does not result in success for an organization. A careful and delicate interplay between choice of strategic applications, appropriate technology, and appropriate organizational responses must be made to attain success, as depicted in Figure 1 [28]. However, no established process or methodology is available for linking strategic applications to appropriate IT and organizational context.

An effective corporation is one that successfully reconciles the problems and opportunities across these three domains. It is important to recognize that no single pattern of interconnection among these three domains is likely to be consistently successful. Thus, one corporation may wish to lead from its technological domain and reconcile the other two domains accordingly. In contrast, another corporation may wish to develop its strategic applications from its product/market choice and develop its technological and organizational capabilities accordingly. It is how the

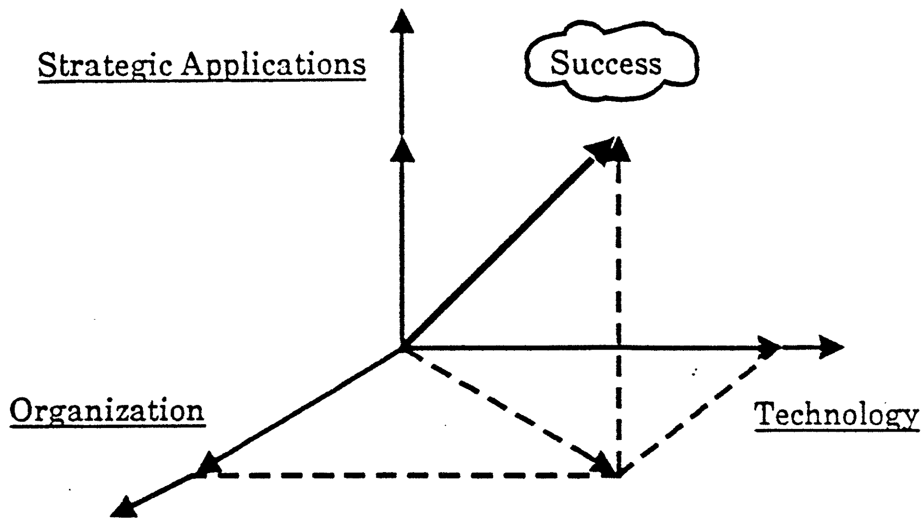


Figure 1 A Strategic Applications, Technology, and OrganizationalResearch Initiative (SATORI)

corporation successfully matches its internal capabilities with the external requirements that determines its level of success in the marketplace.

1.2 Composite Information Systems (CIS)

One important category of strategic IT applications involve inter-corporate linkage (e.g., tying into supplier and/or buyer systems) and/or intra-corporate integration (e.g., tying together disparate functional areas within a firm) of organizational information systems, referred to as *Composite Information Systems* (CIS) hereinafter [13, 21, 27, 28, 29, 34]. The pioneer work on CIS began almost a decade ago [21]. Researchers in the IS field have since then evolved a number of concepts and techniques related to CIS. We summarize some recent work below.

Barrett and Konsynski [3] discussed concepts underlying the growth of inter-organizational information systems (IOS). A classification scheme was presented to examine issues of cost commitment, responsibility, and complexity of the operating environments. Barrett [2] further discussed a range of strategic options and IOS implementations. Cash and Konsynski [8] examined the impact of IOS on corporations. Their work represents a managerial perspective on the development and deployment of CIS.

In linking business and technology planning, Benson and Parker [6, 31] argued that business planning should *drive* the technology planning. *Enterprise-Wide Information Management* (EwIM) grids were proposed to enable practitioners as well academics, to apply the EwIM tools of planning in their enterprise. Many of the IS planning tools, such as *Business Systems Planning* (BSP) and CSF, were mapped onto the grids. The work represents a thrust to articulate issues involved in business and IT at the planning level, eventually evolving into a methodology for linking strategic applications to appropriate IT and organizational context.

In the technical arena, the *MULTIBASE* research project at Computer Corporation of America [11, 14, 22, 37] provided a uniform interface through a single query language and database schema to data in pre-existing, heterogeneous, distributed database management systems (DBMS). The *federated architecture* [18, 23] provides mechanisms for sharing data, for combining information from several components, and for coordinating activities among autonomous components via negotiation. Furthermore, twelve methodologies developed over the past decade for *view integration* and *database integration* have been analyzed and compared by Batini, Lenzerini, and Navathe [4]. The INFOPLEX project at MIT has also been directed towards developing next generation distributed database architectures [26, 27]. In parallel, Hewitt and Agha [1, 19] deal with highly parallel, distributed, *open systems*. The Acore language was developed to facilitate the message passing semantics in open systems. The underlying assumption of their research thrusts is that future IS applications will involve the interaction of systems that have been independently developed and administered.

These research experiences can be very important for implementing CIS with high return on investment, as illustrated below.

2. Strategic CIS Opportunities

Consider the following case study of a major international bank [34]. Currently, three separate database systems, as shown in Figure 2, are being used for

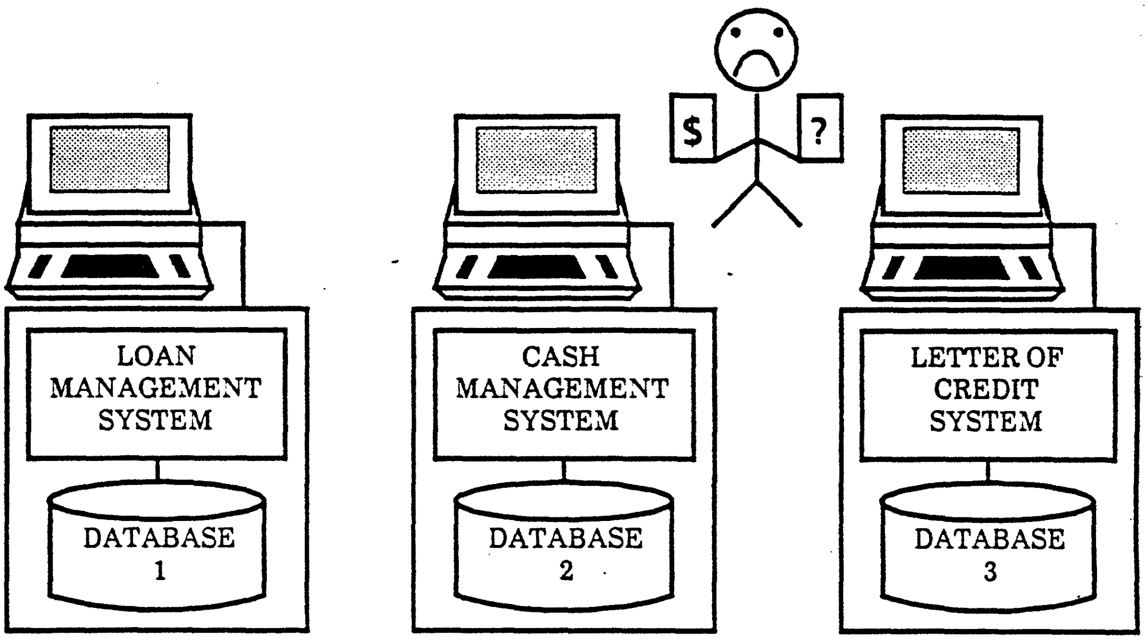


Figure 2 An Electronic Banking System Without Integration

cash management, loan management, and letter-of-credit processing. Suppose a client requests that $20,000 be transferred to another account. If that client's cash balances in the funds transfer system can not cover that transaction, it will be rejected -- even though that client may have a $1,000,000 active letter-of-credit! This rejection, besides being annoying and possibly embarrassing to the client, will require significant effort to correct by manually drawing on the letter-of-credit to cover the funds transfer. If the bank can connect the three separate database systems together to access information in concert, so that funds can be automatically drawn on the letter-of-credit, then product-differentiation will be achieved via the enhanced quality of service and reprocessing costs can be reduced since special manual intervention can be avoided.

Note that the opportunities for CIS are not limited to inter-product integration. Four categories of potential CIS opportunities have been identified: (1) inter-corporate; (2) inter-divisional; (3) inter-product; and (4) inter-model. In reality, a CIS may exploit a combination of these categories. The following subsections exemplify these opportunities.

2.1 Inter-Corporate Applications

Systems in this category involve two or more corporations (e.g., direct connection between production planning system in one company and order entry system in another company). For example, American Hospital Supply (AHS), a manufacturer and distributor of a broad line of products for doctors, laboratories, and hospitals, has since 1976 evolved an order-entry and distribution system that directly links the majority of its customers to AHS computers. Over 4,000 customer sites are linked to the AHS system (i.e., an inter-corporate application). As well as providing the customer with direct access to the AHS order and distribution process, the system supports many customer functions, such as inventory control and summary reports.

The AHS system has been successful because it simplifies the ordering process for customers, reduces costs for both AHS and the customer, and allows AHS to develop and manage pricing incentives to the customer across all product lines. As a result, customer loyalty is high and AHS, which started out as a fairly small company, has gained a significant market share [5].

2.2 Inter-divisional Applications

Systems in this category involve two or more divisions within a firm (e.g., corporate-wide coordinated purchasing). For example, TOYS R US, the number one toy chain in the United States, developed an inter-divisional information system which provides intra-corporate integration of more than 165 toy stores it owns. With the inter-divisional information system as its competitive weapon to keep track of what is selling so it can rapidly replenish fast movers and cut the price of slow movers to make room for the hot items, the corporation captured an impressive eleven percent share in the toy market. Although this information system is not the only reason for the chain-store's rapid growth in the highly competitive industry, it is certainly essential [40]. A CIS capable of accessing all the pertinent databases across divisions can provide timely information to reduce ordering and inventory costs.

2.3 Inter-Product Applications

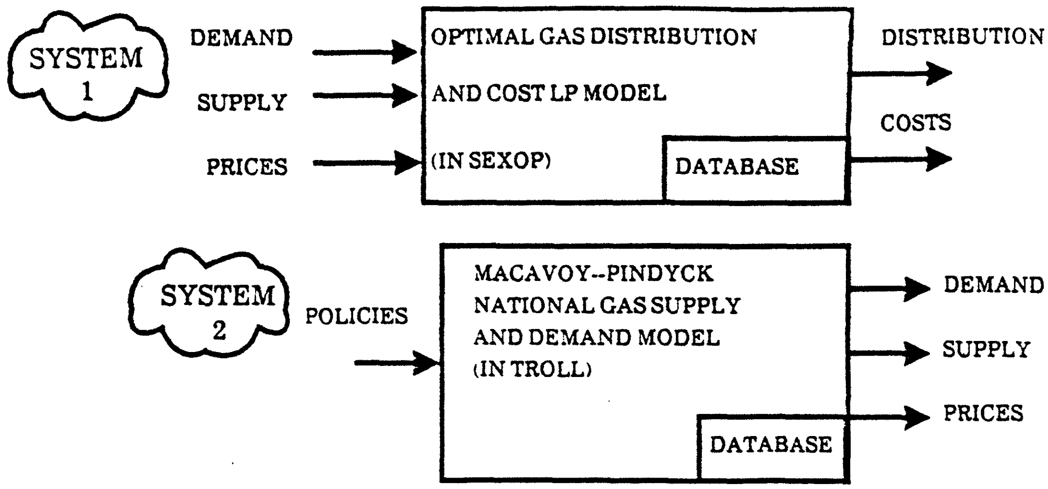
Systems in this category involve the development of sophisticated information services by combining simpler products. As we discussed earlier, the banking example shown in Figure 2 is an inter-product application. As another example, Merrill Lynch announced in 1977 its innovative Cash Management Account (CMA), an IS-based service that provided under one umbrella three appealing services to investors: credit through a standard margin account, cash withdrawal by check or Visa debit card, and automatic investment of cash, dividends, etc., in a Merill-managed money market fund.

Without the IT which provided a complex interface of telecommunication and database management systems, this inter-product application would never have gotten off the ground. With the CMA account, Merrill brought in over 450,000 new accounts, reaped over $60 million a year in fees, and dominated the market for four years. Competition from other financial services organizations did not begin to appear until 1981 [40].

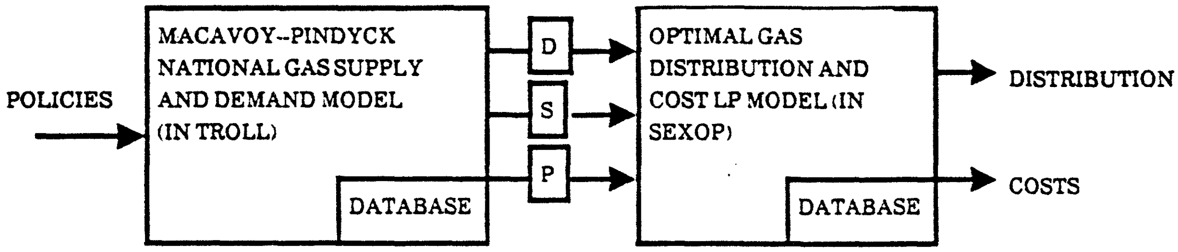
2.4 Inter-Model Applications

Systems in this category involve combining separate models to make more comprehensive models. As part of energy policy analysis research at MIT, the MacAvoy-Pindyck gas model, using the TROLL econometric modeling system, was developed to study the impact of government policies on the demand, supply, and prices of natural gas. To determine the optimal distribution schedule and its cost, the New England Regional Commission was interested in combining its Decision Support System (DSS), using the SEXOP linear programming optimization system, with the MacAvoy-Pindyck model, as shown in Figure 3, to explore the impact of the various government policies on profits. This was a major challenge since the two systems had been developed independently with different tools, languages, and databases.

We have exemplified the strategic opportunities of CIS. It should be emphasized here that the problem is not one involving data alone. Instead, process- or program-related information also must be selectively retrieved to generate meaningful results. Present-day systems are generally inadequate to handle the situation. Newer techniques need to be developed to allow easy, efficient, and intelligent access to information hosted on heterogeneous systems. The following section examines principles of CIS to provide guidelines to the evolution of CIS.



(A) Independent Modeling Systems



(B) Desired Composite Modeling System

Figure 3 An Energy DSS Example

3. Principles of CIS

As pointed out earlier, a composite information system is a system which integrates "independent" systems which may reside within and/or across organizational boundaries. By "independent" we mean systems which are (or were) developed independently, usually by separate groups or organizations. It is crucial to realize that the "independence" of these systems is not necessarily a mistake. It is often driven by needs for division of responsibility, organizational autonomy, and/or differences in objectives. However, it may be important to access these systems in concert for certain purposes.

The traditional approach to system development, not sensitive to the synergistic issue, tends to result in sealed systems as illustrated in Figure 2 and abstracted in Figure 4. The process, model, or tool of system 1 do not communicate

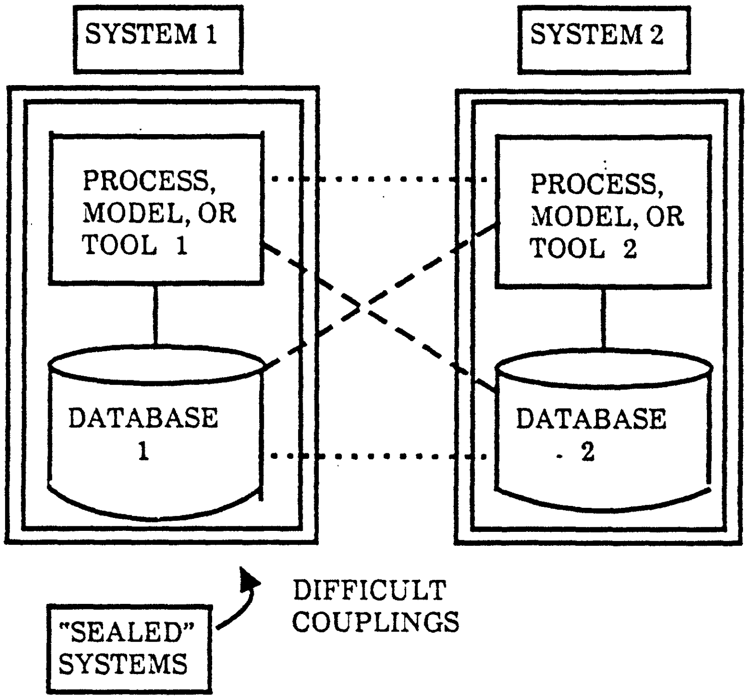


Figure 4 The traditional approach to system development

with those of system 2. Cross-access of algorithms and data under this approach is practically impossible.

The CIS approach facilitates cooperation between systems by following certain principles in the system design process. The essence of this approach is captured in the following four principles: (1) the data separation principle; (2) the tool development principle; (3) the interface composition principle; and (4) the CIS environment principle.

These principles are recognized based on the knowledge accumulated from, among others, the ANSI/SPARC architecture [10], fourth generation languages

(4GL), and experiences we have gained [13, 28, 29, 34] in developing composite information systems. The following subsections delineate these principles.

3.1 The Data Separation Principle

In spirit of the ANSI/SPARC proposal [10], this principle provides a logical separation of the database from processing. In order to separate data from the application processing, it is necessary to employ a process descriptor and a database descriptor. The process descriptor describes the name, the input/output data requirement, and other resource requirements of the processing components. The database descriptor contains information about the data (e.g., data model, schema, access rights) in the database, similar to data dictionaries. These two descriptors can be used by the execution environment to coordinate the interaction between the processing component and the database. Flexibility should be carefully designed into the database so the information in the database can be viewed from different perspectives. This allows the database to be accessed by other systems independently developed and administrated. The important point here is that integration of disparate databases will be possible if the principle is followed since local database interfaces (LDI), as demonstrated in MULTIBASE, can be constructed to access data which would not be available otherwise.

3.2 The Tool Development Principle

This principle advocates the usage of software tools, such as special purpose languages, to facilitate the construction of applications. It is in line with the wisdom in 4GLs which employ a small set of powerful commands which are easier to write and more productive. For instance, TROLL is an econometrics model construction language, and TSP is a time series analysis model construction language. These languages provide more specialized, higher-level primitives than traditional general purpose languages (i.e, they are general purpose tools). By allowing applications to be constructed from the same general purpose tools, inter-application communication protocols, which may be cumbersome to implement in the general purpose languages, can be streamlined.

3.3 The Interface Composition Principle

This principle allows interface mechanisms to be built for data conversion and communication between processing components and databases. Three types of interface mechanisms, as shown in Figure 5, have been identified: BRIDGE, LINK, and SPAN.

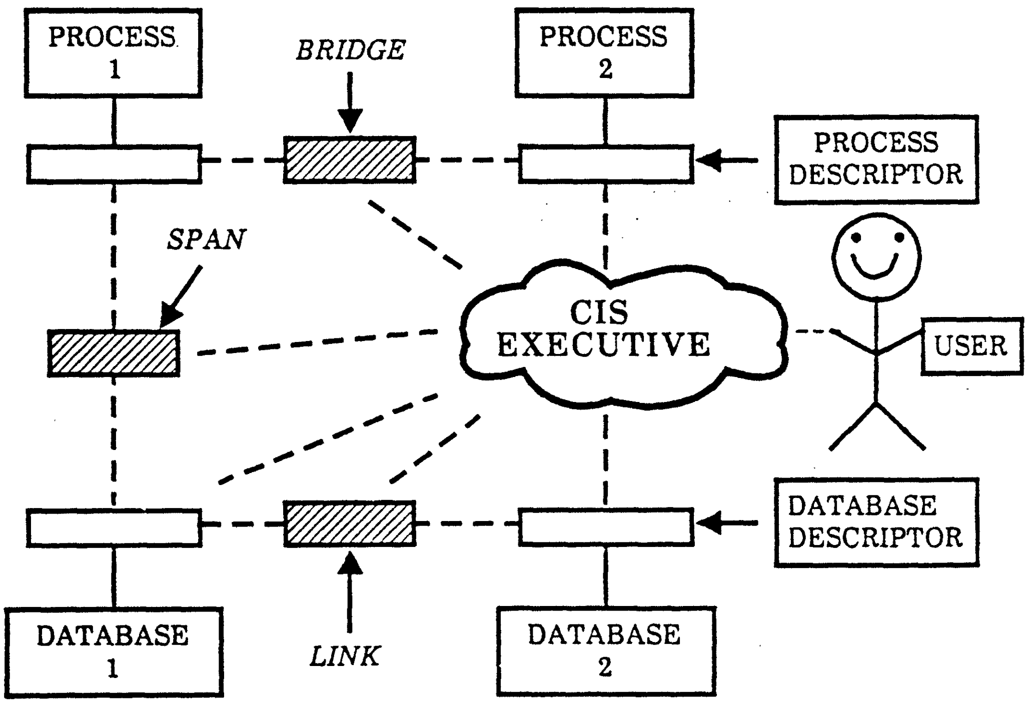


Figure 5 The CIS approach to system development

BRIDGE provides the necessary conversion of arguments to allow invocation of a processing component from another. LINK provides a mapping between two databases with dissimilar types of data models. SPAN converts the data format retrieved from the database into the processing format (and vice versa).

The LDI in *MULTIBASE* and the Export/Import schemata in *Federated Architecture* are two examples of LINK in the sense that they offer the linkage between heterogeneous databases. The *Open Systems* approach is an example of BRIDGE in the sense that processes are encapsulated in actors while actors communicate with one another via messages. The Global Data Manager (GDM) in *MULTIBASE* serves as an example of SPAN in the sense that it insures data from the local databases are syntactically and semantically compatible with what the processing component expects.

The CIS Executive, as shown in Figure 5, mediates the user, BRIDGE, LINK, SPAN, the processing components, and the databases. It directs the request of data from a processing component to the target database, and invokes appropriate interface routines, if necessary, to convert the data, and returns the result to the processing component. It also invokes a processing component on behalf of another.

3.4 The CIS Environment Principle

This principle addresses the need to explicitly allow for the coexistence and usage of a variety of components (e.g., different types of database systems, models, and applications). Moreover, three key conflicting organizational forces were found to have significant impact on the overall environment for CIS: autonomy, evolution, and integration [28]. Tradeoffs must be made among these factors.

It is important to recognize that IS applications are increasingly involving the integration of separate systems that have been developed and administrated independently. Recently, the Center for Information Systems Reserach at MIT's

Sloan School of Management conducted a survey of 31 successful data management efforts in twenty diverse firms [15]. *Information databases* were found to be the most prevalent product in their sample, occuring in 11 (or 35%) of the 31 cases:

> Information databases were defined as a subject area database intended for use by staff analysts and line management. They are "secondary" databases, which periodically draw their contents from operational databases and, sometimes, external sources and often store data in aggregated forms. Significantly, information databases can provide data without requiring major rewrites of current systems. Instead, "bridges" are built from the existing operational systems to provide the appropriate data to the new database.

It would be easy to see that the "bridges" were built based on some of the CIS principles, but not in concert. In contrast, the CIS approach employs the data separation, tool development, and interface composition principles as the building blocks, and explicitly recognizes the CIS environment principle. As such, the opportunities for strategic uses of database technology in organizations can be realized by evolving the existing IT infrastructure in a rapid, yet non-disruptive manner so that local autonomy and global integration can be attained simultaneously. Five stages of evolution of CIS have been developed to facilitate this process: (1) separate systems; (2) the virtual-driver; (3) the logical separation of processing from database; (4) the physical separation of processing from database; and (5) the specialized functional engines. We delineate the process in the following section.

4 Evolving CIS: An Implementation Strategy

4.1 Separate Systems (Stage 1)

The initial stage consists of a set of existing systems that either do not communicate with each other or, more typically, only communicate via human operators, as shown in Figure 2. The processing component and the database component of each system are tightly coupled. The only existing access path is the user interface via the terminal. In order to integrate separate databases in a short period of time to gain timely strategic advantage, the virtual-driver (stage 2) technique can be employed, as discussed below.

4.2 Virtual-Driver (Stage 2)

In the virtual-driver stage, the existing terminal user-interfaces are used to interface existing systems, as shown in Figure 6. *Virtual-drivers* are created which

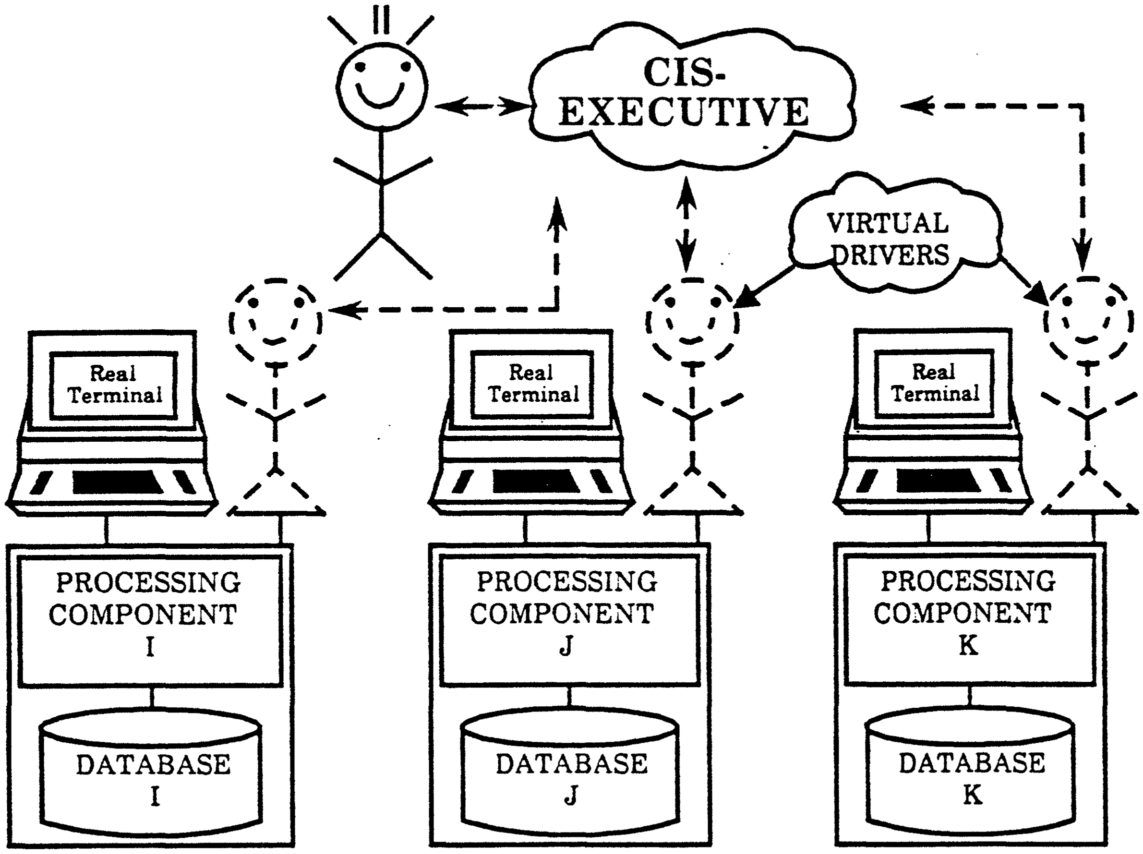


Figure 6 CIS Executive (Stage 2)

are indistinguishable to the system from users of real terminals. Users can still use real terminals to perform their traditional functions. A customer interested in his composite account status may invoke the CIS Executive which mediates the virtual-drivers. The Executive invokes each system (via its virtual-driver) to obtain the necessary information. Incompatibilities between the account data in the two systems are resolved by the Executive which then presents a composite answer to the customer.

It is important not to confuse the concept of virtual-driver with virtual terminal protocols [38]. The virtual-terminal protocols have been invented to try to hide terminal idiosyncrasies from application (i.e., user) programs through mapping of real terminals onto a hypothetical network virtual terminal. In contrast to the narrow mapping of idiosyncrasies among incompatible real terminals, the virtual-driver concept aims at accessing separate databases in concert to formulate composite answers. The mapping in the virtual-driver is twofold: *physical connectivity* in the general ISO/OSI sense [38] followed by *logical connectivity* where reconciliation is made among heterogeneous databases [29].

As an example, we have used UNIX based professional workstations (or personal computers) in several recent applications to link separate systems. Using UNIX as base for the CIS Executive and its CU command to simulate the virtual-driver, it is possible to dial into multiple remote disparate systems. The UNIX workstation appears as a virtual-driver to each of the remote systems. The customer interested in his composite account status invokes a SHELL script which sends the appropriate terminal sequences to each system (via CU), receives the resulting responses (via UNIX "pipes") resolves any incompatibilities between the account data, and finally presents the composite answer to the customer.

The virtual-driver concept is very powerful in connecting separate systems. Very few changes, if any, need to be made to the existing systems, and construction of a CIS-Executive is relatively straight forward. Therefore, a CIS using the virtual-driver approach can be brought up in a relatively short period of time.

As a recent example [25], four banks in the mid-Atlantic states merged, each had developed its own different account status systems (e.g., Burrough, IBM, etc.). To maximize their new market power, it was critical to provide a single coherent account status system rapidly. Using the virtual-driver concept, this was accomplished within a month. This capability is quite important because it can provide functional benefit to the organization quickly and, thereby, sustain upper management support to continue the evolution.

The major drawback of the virtual-driver approach is that it remains difficult to access the databases, which are sealed in each system shown in Figure 6, for purposes not supported through terminal commands. Adding new functions and new types of data is very cumbersome. This leads to the rationale for logical separation (stage 3).

4.3 Logical Separation (Stage 3)

As the organization evolves, one or more of the systems will need to be significantly revised (and/or new systems developed) to meet changing business needs and to keep the systems operationally efficient. At such a point, the CIS principles described in section 3 can be applied. In particular, logical separation of the processing component from its database should be designed into the systems, as shown in Figure 7. By installing a DBMS package, the database activities are offloaded from the processing component of the system. This database is also made available to the CIS Executive through LINK. A dotted line connecting the CIS Executive to the database (see Figure 7) represents new uses of the database by the CIS Executive. Moreover, 4th generation languages such as SQL are employed to increase software productivity. Multiple subsystems may go through this transformation as the system evolves.

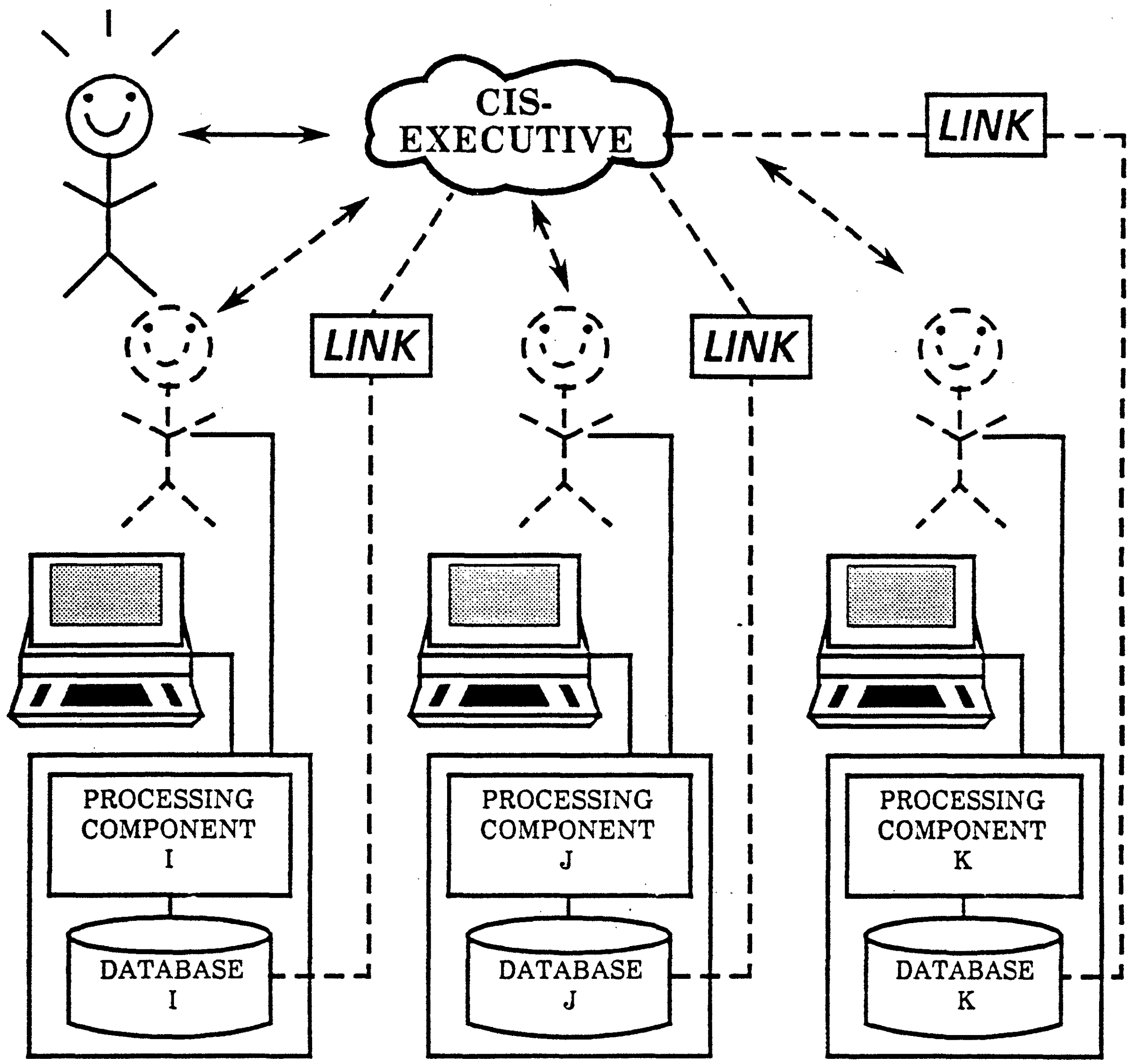


Figure 7 Logical Separation of Processing From Database (Stage 3)

In the twelve methodologies analyzed by Batini, Lenzerini, and Navathe, the basic mechanism used to incorporate disparate databases is comparison, conforming, and integration of the local schemata. As a result, a global conceptual schema is proposed, and tested against the following criteria: (a) completeness and correctness; (b) minimality; and (c) understandability [4]. More recently, Deen, Amin, and Taylor reported [12] PAL (PRECI* Algebric Language) as a facility for data integration in distributed databases. PAL was compared with the DAPLEX work of Dayal and Hwang [11] in solving data integration problems such as name difference, scale and type difference, missing data, conflicting values, semantic difference, and structural difference. Although DAPLEX is semantically richer than the relational approach which underlies PAL, PAL fits neatly with the rest of the query processing stages. These methodologies for view and database integration have offered systematical approaches to implement LINK. However, all the processes require extensive manual effort, and the resulting global schema does not contain explicit knowledge of the assumptions made. Researchers in the IS field have begun to address this issue as well as to include different knowledge sources [20, 29, 39].

Evolution up to this stage has been, in general, software based. Although a CIS application may access multiple DBMS (each of them could be part of a homogeneous distributed DBMS such as ORACLE's SQL*STAR or Relational Technology's INGRES* [17]) in different computers, each computer still holds both the processing and database components. The next stage involves increasing physical separation.

4.4 Physical Separation (Stage 4)

When new computing facility is needed to upgrade the system, two methods are available for partitioning the evolving system: a) migrate a mixture of processing and database components to the new computing facility, and b) partition the processing and database components *physically* and migrate each type of components (i.e, the processing components or the database components) to the appropriate new computing facility. The second method is advocated for the following reason.

On the one hand, one of the often neglected considerations in planning information systems is the need to operate within an environment of "loosely coupled" organizations. The proliferation of personal computers in most organizations is a manifestation of the desire of individual departments or people to control their own computational destiny. On the other hand, it is being rapidly recognized that databases are important resources and the capability to provide timely access can be crucial.

The method of physical separation of processing and database addresses both of these forces by centralizing the databases onto "file servers" or "database servers" that can be accessed by individually controlled (and "owned") application processing elements --- which may range from personal computers to large-scale mainframes, as illustrated in Figure 8. For example, in the case study of a major international bank [13], the databases totally over 22 gigabytes, running on multiple VAX clusters, were segregated from the application processors. Each of the databases was managed by an ORACLE DBMS. Interface enhancements on the database processors made it possible for the databases to be accessed by the application processors via high-speed Ethernets. The application processors were under the direct control of the groups responsible for their operations while the database processors were managed as a shared corporate resource. Furthermore, the separation of the application processing from the data processing paves the way for progressing to the specialized functional-engine stage, as discussed below.

4.5 Specialized Functional Engines (Stage 5)

The increasing demand for information processing capacity has prompted researchers to design large, cost-effective memory systems with rapid access time [16, 27]. One research direction involves database computers which are computers dedicated and optimized for data management, such as INFOPLEX [26]. In the private sector, commercial database machines, such as Britton Lee's IDM 500 and Teradata's DBC 1012 [9] have been introduced successfully. Many of these database computers have adopted highly-parallel, multi-processing architectures to cope with the requirement of high throughput, high reliability, and large storage capacity.

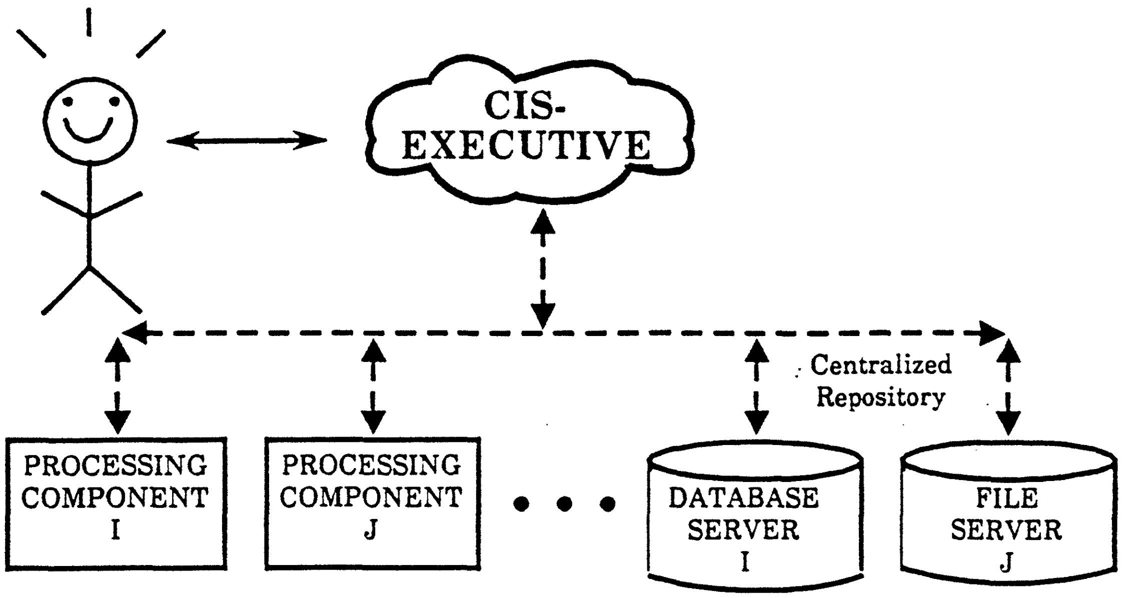


Figure 8 Physical Separation of Processing From Database (Stage 4)

Specialization enables the database computer to handle search, retrieval, and storage of large volume of data more effectively, to provide for adequate capacity to perform complicated data restructuring and mapping, and to enforce security and integrity constraints.

Assuming that an organization has progressed to stage 4, as the technology for database computers continues to mature, the organization can easily upgrade system capacity by migrating the database management tasks performed on a conventional computer to a database computer. Meanwhile, proliferation of professional workstations and personal computers will continue to offload many processing tasks currently performed on a centralized computer. A picture of information systems will emerge as depicted in Figure 9.

Many of the tasks performed by the CIS Executive could be migrated to the database management system or the database computer, such as view mapping, data format conversion, and report generation. These features simplify the task of the CIS Executive which now may reside in the professional workstation or personal

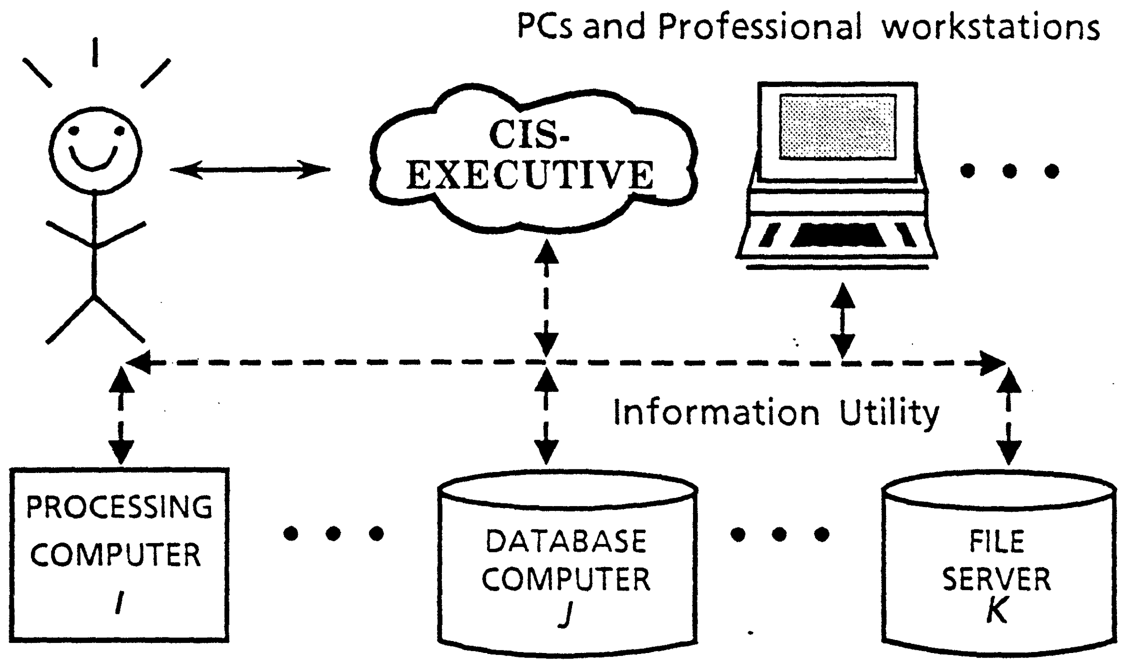


Figure 9 Special Engines for Processing, Database, and CIS (Stage 5)

computer to coordinate access to the resources (processing and database) of the network as well as to mediate steps of internal processing.

The database managed by specialized database computers and controlled by information system specialists of various sub-units of an organization constitute the *information utility* [24]. The end-users, via their desktop computers, access the information utility for data that is either directly usable, usable after further processing by some processing nodes in the network, or usable after further processing by the desktop computer. The CIS at this stage becomes part of the organization's infrastructure to facilitate strategic goals.

5. Concluding Remarks

There are enormous opportunities for businesses to gain competitive advantage through inter-corporate, inter-divisional, inter-product, and inter-model applications. These opportunities for strategic uses of database technology in organizations are often blocked by the difficulties of evolving the existing IT infrastructure in a rapid, yet non-disruptive manner. We have identified four principles and five stages for the evolution of CIS. This five-stage evolutionary process has been found to be effective in overcoming this problem, especially in an autonomous, evolutionary, and integrative CIS environment. These results have provided a foundation for the study of even more advanced applications and technologies to support CIS.

Acknowledgements

Work reported herein has been supported in part by the Department of Transportation's Transportation System Center, the U.S. Air Force, the Center for Management of Information at the University of Arizona, and Citibank. The authors also wish to thank the anonymous referees for their valuable comments on an earlier version of this paper.

Received in June 1987; revised in September 1987.

References

1. Agha, G. ACTORS: A Model of Concurrent Computation Distributed Systems. The MIT Press, Cambridge, Mass. (1986), Chap. 3 -7.

2. Barrett, S. Strategic alternatives and Inter-Organizational Systems implementations: an overview. Journal of MIS, Vol. III, No. 3 (Winter 1986-87), pp. 3-16.

3. Barrett, S., and Konsynski, B.K. Inter-organization information sharing systems. MIS Quarterly, Special Issue (1982), pp. 93-104.

4. Batini, C., Lenzerini, M, and Navathe, S.B. A comparative analysis of methodologies for database schema integration. ACM Computing Surveys, Vol. 18, No. 4, (December 1986) pp. 323-364.

5. Benjamin, R.I., Rockart, J.F., Scott Morton, M.S., and Wyman, J. Information technology: a strategic opportunity. Sloan Management Review, Vol. 25, No. 3, (Spring 1985), p. 3-10.

6. Benson, R.J. and Parker, M. M. Enterprise-wide information management: an introduction to the concepts. IBM Los Angeles Scientific Center, G320-2768, (May 1985).

7. Beath, C.M. and Ives, B. Competitive information systems in support of pricing. MIS Quarterly, (March 1986), pp. 85-96.

8. Cash, J. and Konsynski, B.. IS redraws competitive boundaries. Harvard Business Review, March-April, 1985.

9. Computerworld Database machine's appeal rising. (May 20, 1985), pp. W-3.

10. Date, C.J. An Introduction to Database Systems. 3rd Ed., Addison-Wesley, (1981), pp. 17-32.

11. Dayal, U. and Hwang, H.Y. View integration and generalization for database integration in a multidatabase system. IEEE Transactions on Software Engineering, Vol. SE-10, No. 6, (November 1984).

12. Deen, S. M., Amin, R.R., and Taylor M.C. Data integration in distributed databases. IEEE Transactions on Software Engineering, Vol. SE-13, No. 7, (July 1987) pp. 860-864.

13. Frank, W.F., Madnick, S.E., and Wang, Y.R. A conceptual model for integrated autonomous processing: an international bank's experience. Proceedings of the 8th International Conference on Information Systems (December 1987).

14. Goldhirsch, D., Landers, T., Rosenberg, R., and Yedwab, L. MULTIBASE: a system administrator's guide. Computer Corporation of America (November 1984).

15. Goodhue, D., Quillard, J. and Rockart J.F. Managing the data resource: a contingency perspective. CISR WP No. 150, Sloan WP No. 1859-87, 90s WP No.

87-032, Center for Information Systems Research, Sloan School of Management, MIT (January 1987).

16. Goyal, A. and Agerwala, T. Performance analysis of future shared storage systems. IBM Journal of Research and Development (January 1984), p. 126-138.

17. Gref. R.W. Distributed database systems: a comparison between ORACLE and INGRES. Master's Thesis, Sloan School of Management, MIT, (May 1987).

18. Heimbigner, D. and Mcleod, D. A federated architecture for information management. ACM Transactions on Office Information Systems, Vol. 3, No. 3, (July 1985), pp. 253-278.

19. Hewitt, C. E. Office Are Open Systems. ACM Transactions on Office Information Systems, Vol. 4, No. 3, (July 1986), pp. 271-287.

20. Kerschberg, L. Ed. Expert Database Systems, Proceedings from the First International Workshop. The Benjamin/Cummings Publishing Company (1986).

21. Lam, C.Y. and Madnick, S.E., Composite Information Systems - a new concept in information systems. CISR WP# 35, Sloan School of Management, MIT, (May 1978).

22. Lander , T. and Rosenberg, R. An overview of MULTIBASE. CCA Technical Report, (1983).

23. Lyngbaek, P. and McLeod D. An approach to object sharing in distributed database systems. The Proceedings of the 9th International Conf. on VLDB, (October, 1983).

24. Madnick, S. E., Trends in computers and computing: the information utility. Science, Vol. 185, (March 1977), pp. 1191-1199.

25. Madnick, S. Merging banking systems: a Composite Information Systems approach. Internal Documentation, Cambridge Institute (219 Vassar Street, Cambridge, MA 02139), (June 1986).

26. Madnick, S. E. and Hsu. INFOPLEX: research in a high-performance database computer. IEEE Database Engineering, Vol. 9, No. 1, (March 1986).

27. Madnick, S. E. and Wang, Y. R., Modeling the INFOPLEX database computer: a multiprocessor systems with unbalanced flows. Proceedings of the 6-th Advanced Database Symposium (August 1986), pp. 85-93.

28. Madnick, S.E. and Wang, Y.R. A framework of Composite Information Systems for strategic advantage. Proceedings of the Twenty-first Annual Hawaii International Conference on System Sciences, (January 1988).

29. Madnick, S.E. and Wang, Y.R. Integrating disparate databases for composite answers. Proceedings of the Twenty-first Annual Hawaii International Conference on System Sciences, (January 1988).

30. McFarlan, F.W. and McKenney, J.L. Corporate Information Systems Management: The Issues Facing Senior Executives. R. D. Irwin, Inc., (1983).

31. Parker, M.M. Selected Papers on Enterprise-wide information management. IBM Los Angeles Scientific Center, G320-2769, (March 1985).

32. Porter, M. and Millar, V.E., How information gives you competitive advantages. Harvard Business Review (July-August 1985) p. 149-160.

33. Porter, M. E. Competitive Advantage. Free Press, (1985).

34. Rincon, M. N., Integrating systems for financial institutions services using Composite Information Systems. Master's Thesis, Sloan School of Management, MIT, (May 1987).

35. Rockart, J.F. Chief Executives Define Their Own Data Needs. Harvard Business Review, March-April, 1979.

36. Rockart, J.F. and Scott Morton, M.S., Implications of changes in information technology for corporate strategy. Interfaces, Vol. 14, No. 1 (January-February, 1984), pp. 84-95.

37. Smith, J.M. et al, MULTIBASE - integrating heterogeneous distributed database systems. National Computer Conference (1981), pp. 487-499.

38. Tanenbaum, A.S. Network protocols. Computing Surveys, Vol. 13, No. 4, (December 1981), pp. 453-489.

39. Wiederhold, G. et al. KSYS: an architecture for integrating databases and knowledge bases. Computer Science Department, Stanford University, (May 1987).

40. Wiseman, C. and MacMillan, I.C. Creating competitive weapons from information systems. Journal of Business Strategy, (Fall 1984), pp. 42-49.