1-1-1973

Data administration.

Fred Francis Newpeck
University of Massachusetts Amherst

DATA ADMINISTRATION

A Dissertation Presented

By

FRED FRANCIS NEWPECK

Submitted to the Graduate School of the
University of Massachusetts in partial
fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

June 1973

Major Subject Business Administration

DATA ADMINISTRATION

A Dissertation

By

FRED FRANCIS NEWPECK

Approved as to style and content by:

Van Court Hare, Jr., Chairman of Committee
(Dept. of Management)

John T. Conlon, Director of Graduate Studies
(Associate Dean, S.B.A.)

George B. Simmons, Member
(Dept. of Management)

John G. Burch, Jr., Member
(Dept. of Accounting)

Hugh J. Miser, Member
(Dept. of Industrial Engineering &
Operations Research)

(Month) (Year)

ADKNOWLEDGEMENT

I would like to express my sincere gratitude to Professor Van Court Hare, Jr., my dissertation committee chairman and Ph.D. program advisor. His personal involvement and precisely-timed direction throughout the Ph.D. program and the writing of this dissertation was instrumental in the completion of this work. My thanks also go to Professor George B. Simmons, who always put everything in proper prespective, and in addition shares my enthusiasm for computing machinery and APL. Thanks also go to Professors John G. Burch, Jr. and Hugh J. Miser, for their assistance and personal involvement. In addition I am indebted to Mrs. Vesta Powers for her expertise and patience in typing the final copy.

ABSTRACT

The topic of this thesis, Data Administration, covers in total that organizational function concerned with the technology of data processing in large-scale data systems as well as the uses to which a system may be put. Specifically, however, the text is restricted to consideration of Data Administration Technology, first to limit the scope of the work, and, second, because it is the author's thesis that the technological body of knowledge required to perform the data administration function has in the past been insufficiently stressed so that electronic data administration appears as a librarian's task, which it is not. Moreover, although the details of application and organizational structure may vary widely in practical data administration, certain technical patterns of operation are common and may be generalized, whereas the specifics usually belong to each given case, further suggesting that technological requirements and capabilities provide a more universal starting point for an extended study of this new organizational function. Coverage of material consequently includes: Data Input Systems, Data Communication Systems, Data Base Management Systems, and Monitors and Measures of System Effectiveness. Extensions of the material are indicated for subsequent studies.

FORWARD

In April, 1971, a document entitled "Data Base Task Group Report" appeared as the sequel to previous historical efforts to develop and standardize a business computer language. [1]

Composed of both business and computer equipment specialists, the report's authors sought to extend previous efforts which had culminated in the specifications for the now widely used computer programming language known as COBOL (Common Business Oriented Language). The merits and disadvantages of this language may be found elsewhere. [2] Briefly, however, although COBOL provided English-language programming capability--which was in that sense self-documenting and also in that sense capable of wider understanding by users not expert in computer technology--early specifications of the language were severely constrained by computer memory and associated device technology of the 1960's. Resulting applications, confined to rigid and inflexible file structures, usually sequential or at most indexed-sequential, [3] proved inadequate as years passed and as applications grew in complexity and interdependence.

So, proposed improvements in the COBOL language set forth by the Task Group were drastic by older standards: not only were file structures to be flexible and in many alternate forms, but also many modes of data input and data output were to be available. Moreover, users of the data processing system from the least to the most knowledgeable were to be accommodated by a series of internal file and system control features which would make use of the data system simple--for the novice and expert alike--while at the same time maintaining some semblence of economic, organizational, and technical reality.

The objectives of the Task Group were noble not only in the technical appreciation of the problem, as discussed hereafter, but also in the general philosophy which inevitably follows from serious consideration of the need for stable change.

For example, in any given organization, we know that over time the detail needs for data will shift. Similarly, the technology and economics available to service such needs will shift, too.

Knowing this, and realizing full well that added options and added flexibility would require some form of management, the Task Group proposed--somewhat in passing to be sure--a new organizational function, that of Data Base Administrator.

Without going into the Task Group's specific details on the subject, such an administrator, or function was to take care of whatever "tuning" or adjustment might be needed over time to make sure the data system, particularly the files or data base included in it, were always up to snuff. The general impression when one reads the Task Group's work is that the DBA should be some form of electronic librarian. My thesis is that the impression is not so, the Data Base Administrator has broad technical responsibilities pertaining to the control of data in the organization. His technical responsibilities begin with the data source and extend to the user, following the path of data through the telecommunications system and the data base to the user.

Since his data responsibility extends beyond the data base, Data Administrator is more descriptive of his broad duties as data systems engineer. Aerospace companies identify the system engineer as that specialized function charged with the responsibility of bringing cost-effective balance to the total system while meeting system objectives. System engineering continues throughout the life cycle of the product, it involves selecting sensors, actuators and the control criteria on a continual basis as technology changes or the system is asked to meet a new environment.

The Data Administrator in this dissertation resembles the aerospace system engineer as he selects input sensors, data base management system actuators and the data control criteria to meet stated objectives of data quality while maintaining a technological cost-effectiveness balance, as the user environment and technology changes.

The Data Administration function includes broad organizational and human implications which are highly dependent upon wide ranging details of computer-based information system application, management objectives, and organizational and environmental structure. However, certain technical patterns of operation are common and may be generalized, forming a universal starting point for an extended study of this new organizational function. This dissertation identifies and evaluates the technological basis for Data Administration.

FOOTNOTES

1. CODASYL Data Base Task Group, April 1971 Report, (New York: Association for Computing Machinery).

2. H.T. Hicks, Jr., "ANSI COBOL," Datamation, (November 1, 1970), pp. 32-36.

3. G.G. Dodd, "Elements of Management Systems," Computing Surveys, Vol. 1, No. 2, (June, 1969), pp. 119-120, 128.

TABLE OF CONTENTS

LIST OF FIGURES

C H A P T E R I

THE EVOLVING DATA ADMINISTRATION FUNCTION

Evolution of the Data Base Concept

The increasing size, diversity and complexity of organizations argues for a more systematic approach to monitoring operations. Reporting requirements become more extensive and planning more difficult. Managers must be prepared to respond to an ever-broadening range of internal and external situations, subject to ever-narrowing time constraints. To further complicate matters, the sheer volume of operations has increased enormously in most organizations, while the cost of labor has skyrocketed.

The information systems available to those in charge of operating and directing organizational activities must keep pace with these trends. Information systems have always existed in organizations, and with the advent of computer techniques their operation has become a highly technical subject. The result has been that information processing activities have become an area of specialization within organizations. As more and more of the information processing is performed within a machine, the need for technical expertise becomes ever more apparent. Appendix A presents a summary of information system evolution and the design concepts necessary to an understanding of information system design and implementation.

The final phase in the evolutionary development of a computer-based information system implements the data based information system implements the data base concept. The concept is the direct result of the need to meet an environment of increasing complexity and rapid change with scarce and limited human resources. New information systems can be implemented more rapidly and efficiently if the data gathering task can be eliminated. The data base concept does precisely this by storing all relevant internally and externally generated data in machine readable form which can be accessed by any application program. Information processing procedure is independent of data structure and the information system and the data base can grow independent of one another.

Data Administrator Resolves Conflict Resulting From Data Base Implementation

The data base concept because of its efficiencies makes the expansion of computer-based information systems to tactical and planning levels cost feasible. However, the sharing of common data among diverse applications leads to contention and the need for compromise. The Data Administrator function is the human activity of mediating disparate user data needs.[1] The CMSAG System Report Group says that the Data Administrator is the person or group that plans, coordinates, and controls the use of all data and information resources available so that individual users can be assured that the right information will be made available to the right employee or process at the right time. (CSMAG is the Construction Management System Action Group composed of 17 U.S. and Canadian Gas and Electric Utilities.)[2]

A Kaleidoscope of Data Administrator Responsibilities

R.G. Canning analyzed the Data Administration function in the November, 1973, EDP Analyzer. He looked at the function as implemented at Pillsbury, Welsco Data Corporation and Pacific Gas and Electric as well as examining a 1970-71 survey of 17 companies identifying the function. He concluded that each organization saw the problem in different terms, depending on the needs of the particular organization.[3] A review of the current literature reveals that different authors look upon the Data Administrator function in a variety of ways. There is as yet no consensus on the scope and boundaries of the Data Administrator function.[4] A sample of the diversity of viewpoints appear as Appendix B.

Data is a Corporate Resource

Though the Data Administrator function has not yet clearly evolved, data in itself is becoming accepted as a corporate resource on a par with capital and human resources. In a recent working paper Richard L. Nolan surveys ten companies concerning the data base issue and the Data Administrator's responsibilities. He found that comments on what the responsibilities should be tended to treat data as a resource, much broader than just computer readable data. The responsibilities of the Data Administrator included activities to advise and help management effectively use data. The comments on what the responsibilities actually are tended to treat data as computer-readable data only. He concludes that this dichotomy is the result of the Data Administration function being carved out of the general management function. [5]

Data Administration at its First Evolutionary Level

If this broad view of data responsibility is valid, why has it not appeared in practice? Currently, the Data Administrator is little more than a clerk. At a recent SIGFIDET workshop I attended one discussant lamented on finding someone interested in performing the tedious task. [6] Obviously, the Data Administrator function is at its machine language or unit record equipment level in its evolutionary development.

Currently, the literature stresses the Data Administrator's role as it pertains to the data base. Such questions as data accuracy, integrity and security are related to the data base alone. A major problem is the unconstrained problem of file design, which is still an art in the data processing field.[7] Another major problem is lack of a body of knowledge which will directly indicate that a portion of the data base is deteriorating either in efficiency or integrity.[8]

Promethean Computer and Telecommunication Technology

The reason for the paucity of knowledge concerning computer-based information systems is the promethean nature of computer and telecommunication technology. Just as Prometheus stole fire from heaven and gave it to man, and then man had to learn to use and control it, so it is with computer technology. The first computer-based multilevel information system did not appear until around 1968,[9] and the data communication links necessary to control geographically dispersed situations have only recently become available. In 1956 transatlantic voice communication became popular with the opening of a high quality 36 channel cable. Today, eight satellites provide 25,000 voice channels with growth to 250,000 channels predicted by 1982.[10]

That computer technology is a young science is reflected in the literature. The number of periodical titles are doubling every 4 to 6 years as compared with a doubling period of 10 to 15 years for science and technology as a whole.[11] Other indications of a young, ill-structured technology is that the literature is widely scattered over many journals[12] with a low 8 references per article average compared to the 16 or more average for a hard science.[13] Because Data Administration has only recently begun to be recognized as a separate need, the literature on the subject is even more ill-structured than for the information processing field as a whole.

Data Responsibility Extends from Input Sensor to User Application

Data accuracy and integrity. The reason the Data Administrator role is tied so closely to the data base is that currently its management is the central problem. My view is that data resource management springs from the data base, but is much broader than the data base. Data responsibility begins at the data sensor and extends to user application. The user must have one person or function he can turn to for assurance that the data he is using meets his criteria for accuracy, integrity and security. To give this assurance the Data Administrator must know how the data was sensed and what error tolerances were allowed, he must know how the data was transmitted to the data base and what precautions were taken in its transmission to limit data degradation, and finally he must administer the data once it is in the data base to maintain its reliability. Because data specialization is much broader than the data base, the title Data Administrator is more descriptive of the function than Data

Base Administrator.

Data response. Getting accurate, meaningful data to the user is one half of the problem. The other half requires that the data be sufficiently prompt to affect user control actions. This is especially true at the operational level where response times can be measured in seconds, minutes or hours. Tactical control response requirements extend to days or weeks and strategic control requirements extend to months or years. Hence, the data base must be structured to maximize operational data response. In most present systems the data base is centralized and the users in close proximity, in this instance data base structure is the single most important factor affecting response. In reservation systems the users are geographically dispersed, and the response of the input and output sensor and the response of the communication link become important. In future systems the data base may be geographically distributed, the communication link becoming critical to response requirements.

Data flow monitoring. The point is that the physical information system structure varies to match the firm's information structure. The result is a faster responding system which incorporates telecommunications technology and remote intelligence. This technology is just as much a part of the total Data Science as is Data Base Technology, and the Data Administrator is the control agent. Control of the data input system, the telecommunication system and the data base presupposes a means of control. The emerging monitoring function is the control mechanism, the means to measure data flow, and therefore, should be the responsibility of the Data Administrator.

A Body of Knowledge for Data Administration

This dissertation will examine the extent of Data Administrator responsibility for technology. It will not rigorously examine human relationships, data source/use locations and technology or the Data Administrator position in the organization. These facets of data responsibility are highly dependent upon the organization's structure; however, a common stream of data technology runs through all structural variations, this technology will be examined in detail and the most likely application environment depicted.

Historically, Data Administration began with the data base, this technology will be examined first, logically followed by the technologies of data input, telecommunications and computer-based hardware/software system monitors. The scope and limits of Data Administrator technical control and responsibility will be delineated. The result will be the identification of a body of knowledge pertinent to Data Administration.

FOOTNOTES

1. CODASYL Data Base Task Group Report, (New York: Association for Computing Machinery), April, 1971, p. 22).

2. "Data Administration Working Paper," (April 21, 1973), p. 3, CSMAG Data Management Project, J.N.R. Bryan, Chairperson, Florida Power and Light Company, Box 3100, Miami, Florida 33169.

3. R.G. Canning, "The Data Administrator Function," EDP Analyzer, Vol. 10, No. 11 (November, 1972), p. 3, 6.

4. Ibid., p. 12.

5. R.L. Nolan, "Managing the Data Resource: The Organization Issue," Computer-Based Information System Working Paper #2, Revision 3 (Boston: Graduate School of Business Administration, Harvard University, May 7, 1973, p. 16.

6. "The Information System and Data Base Management," SIGFIDET Work Session, New York City, March 29, 1973.

7. J.P. Fry, Personal Correspondence, Department of Industrial and Operations Engineering, College of Engineering, The University of Michigan, Ann Arbor, Michigan 48104.

8. J.K. Lyon, "The Role of the Data Base Administrator," Data Base, Vol. 3, No. 4 (Winter, 1971), p. 12.

9. R.I. Benjamin, "A Generational Prespective of Information Systems Development," Communications of the ACM, Vol. 15, No. 7 (July, 1972), pp. 640-643.

10. H. Busignies, "Communication Channels," Scientific American (September, 1972), p. 99.

11. A. Pritchard, A Guide to Computer Literature (Hampden, Conn.: Shoe String Press, Inc.), 1972, p. 7.

12. Ibid., p. 13.

13. Ibid., p. 14.

C H A P T E R II

DATA BASE ADMINISTRATION

Introduction

Historically, Data Administration evolved from the necessity to resolve data conflicts among disparate users of a common data base. Currently, Data Base Administration is becoming only one facet of Data Administration. This chapter discusses Data Administrator responsibilities and considerations as they relate to the data base, to include, data sources, data base content control, data structure, physical data structures, data security and data management systems.

Sources of Data

Internal data sources. Data sources may be either internal or external to the organization. Internal sources provide the bulk of the operational level data. Tactical and strategic levels utilize increasing amounts of externally generated data. Appendix A details how to identify and coordinate internal sources of data. First, the operational level, mainline information flow, is identified and modularized in such a way that data and processes are independent of one another. Module boundaries then locate data sources.

External data sources. External sources of data are more difficult to identify as much of the external data needed at tactical and strategic levels is not used on a continuing basis. Data collection on a one-shot basis is very expensive and many times prohibits analytical model building at tactical and strategic levels because the cost of data sensing is not worth the benefits. Information utilities and availability of inexpensive computer readable data from government agencies, industry groups, and other external sources is changing the cost-benefit picture.

For example, in Canada a new geocoding system (geographical referenced data storage and retrieval system) will provide census data to anyone for any part of Canada down to an area as small as a few blocks along one side of a street. Information as to age, sex, marital status, mother tongue, land classification, and whether property is owned or rented is available for as little as $14 with additional charges for populations covered and for each extra variable included.

The U.S. National Technical Information Source (NTIS) of the Department of Commerce provides literature search of government-funded reports for $25 per question. Magnetic tape files in highly specialized categories are also available as are census files.

Another source of data is the New York Times morgue, currently the world's largest general information retrieval system. The contents of 3500 filing cabinet drawers filled with 20 million clippings from 60 different publications covering 80,000 subjects and 1.3 million people are on-line, available to anyone on a fee basis.[1]

R.L. Laska, Senior Editor at Computer Decisions describes the user of various external sources.[2]

> The marketing director of a firm considering selling in the European market can find out the latest tariff situation in his field from the Times data bank. Then he can tie in with the economic modeling service of Chase Econometrics, Data Resources, Rapidata or others to determine the anticipated growth in his economic sector. Then he can access Sumstat, the Department of Commerce' proposed statistics file, to get more specific information on relevant trade patterns. Then switching to his in-house data base and accessing the corporate model, he can determine the impact of entrance expenses for the European market on future domestic market plans.

The effect of utilizing external data from information utilities at reasonable prices will bring a whole new capability to managers. Their decisions will more likely be based on solid information and less on instinct. The Data Administrator will be responsible for knowing the location of external data and its reliability as well as providing for its availability to decision makers.

Data Base Content

Boundaries of data base content. An important decision to be made by the Data Administrator is what data will reside in the data base and what should be the data retrieval response. These two decisions are closely related because the data retrieval response requirements, access frequency and importance to organizational control, determine whether the data will be stored on-line or off-line. If the data is critical and accessed frequently, it will be stored on-line. If the data is infrequently accessed and not critical to day-to-day operations, perhaps it can be stored off-line. The boundaries of on-line data base content must be established; otherwise, the data base will contain dead space, causing unnecessary collisions between working data as they expand and contract as the condition they describe changes.

An example of data files expanding and contracting is the existence of raw material as it enters the production process. As the raw material passes through the system its identity changes as it approaches the finished product. Subassemblies are manufactured combining raw materials with other raw materials and subassemblies; all the while the condition of the production floor are contained in the data base files, which expand and contract with changing conditions. If unused data fills the physical storage device, there is less room for oscillations in file size. The files may become fragmented and the data base may have to be reorganized; therefore, it is more efficient to remove little used, non-critical data from the data base. Frequency of data element access statistics, can assist the Data Administrator by indicating infrequently used data. The data dictionary to be discussed later provides the means to measure access frequency.

Identify redundant data. Other means of reducing data base content is to identify data redundancies. A data element may be the same or have different names. Multivariate statistical correlation techniques could aid the Data Administrator in this area. Other redundancies may develop such as when the same or similar names are used for the same data-item or for almost the same data-item. Similar discrepancies can also occur in data-item mode, length, etc.[3]

Several data dictionary software packages are on the market today to assist the Data Administrator in resolving data conflicts and insuring that each data item is stored only once. Western Electric Company initially feared that they might have to identify as many as 30,000 different data items, but when the redundancies were eliminated and some data was derived from other basic data-items it was found that only 2500 data-items needed identification.[4] Correlation techniques are useful in identifying redundant and derived data. For example, if a list of birth dates were compared to a list of ages the canonical redundancy coefficient would indicate the data was the same (one could be derived from the other).[5]

Data compression. Another means of saving space is data compression in which case additional processing is traded for reduced storage space, a reduced transmission load over costly circuits and a type of simple encryption. One data compression software package developed by DeMaine and Springer, of Pennsylvania State University called COPAK, stores compressed data as a bit string, with self-defining data at the head of the string.[6]

DeMaine and Springer claim compression of 80% to 85% for numeric data and 45% to 75% for alphanumeric data. Typical machine overhead is suggested by a 57% compression of a personnel file, 1300 byte segments were compressed in 130 milliseconds on an IBM 360/65.[7] Compression factors in the 50% to 60% range appear elsewhere in the literature indicating that file size reduction is indeed significant.[8]

The technique employed by DeMaine and Springer is as follows: Numeric data is compressed by (1) truncation of fixed and floating point numbers, (2) differencing between numbers, (3) scanning for repetitions of the same integers, or (4) packing data into double words. If no saving is achieved data is stored in its raw form. Alphanumeric data is compressed by scanning for repetitive bit patterns, starting with 96 bit strings, which are reduced 8 bits at a time in search of patterns. When a repetitive pattern is found it is assigned a code and entered in a decoding table and in the records. Once a pattern table has been constructed for a reasonable sample, analysis proceeds at a 200 to 500 increase in speed.[9]

chores. It automates data base definition and standards control.

One example of a commercial data dictionary is the Data Catalogue, a software product of Synergetics Corporation. The Data Catalogue generates the following reports:[11]

Currently available:

CATALOGUE REPORT -- shows the definition of format and content for all data entries in the catalogue system.

INDEX REPORT -- shows by catalogue name, the page on the catalogue listing which contains the definition for the data entry indexed.

CROSS REFERENCE REPORT -- the most complex report. The report cross references all the elementary data-items, group items, records, and files.

Under development:

DATA BASE STRUCTURE REPORT -- shows pictorial representations of the various structures in the data base.

ELEMENT AND SEGMENT SECURITY REPORT -- provides a list of data items and records having various levels of confidentiality as defined in the Data Catalogue. This report aids data base design by indicating which elements and records should be stored together for better security.

FREQUENCY OF USAGE ANALYSIS REPORT -- provides a report of data items described in the Data Catalogue by levels of access frequency. The report is useful in spotting high access data-items in segments of mostly low access elements.

DATA BASE ANALYSIS REPORT -- documents the structures, pointers, keys, etc.

Such data dictionary software is a convenient tool for communicating with analysts and maintaining data base standards. Such systems perform many of the boring, menial tasks of the Data Administrator.

Data dictionaries assist the manager in attaining the real objective of data management, which is not the elimination of redundant data, but the control of data sources. Often it is desireable to maintain and process duplicate data because of constraints imposed by data structures and hardware constraints. The data dictionary helps prevent accumulation of truly redundant data by presenting to the Data Administrator, a clear, unequivocal description of the data resources and the uses to which data is put. The data dictionary presents the relevant evidence upon which the method of data source control is selected.

Data Structures

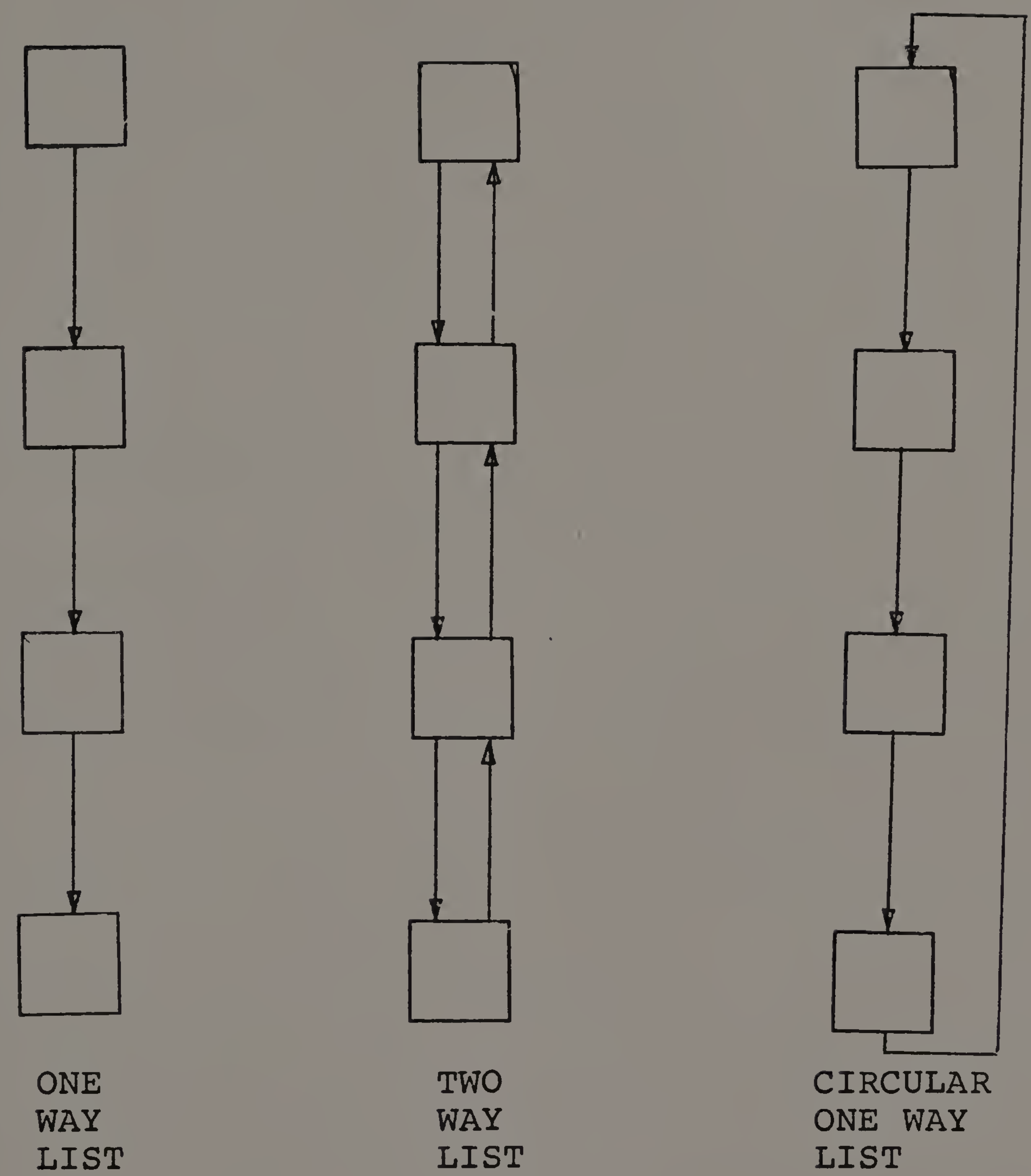
Introduction. Once the data to be stored in the data base is identified and redundancies eliminated, the data linkage structure, called the data base (storage) structure, must be designated. Ideally, the firm's natural data structure, by category, mirrors the data base structure. However, hardware constraints, especially the discreteness of hardware storage devices, results in the data base physical structure being different from the firm's data structure. T.W. Olle of RCA calls data structure, that structure the user must be aware of in order to process the data base with the facilities at his disposal.[12] Thus data structure refers to that structure evident in the data base management system, data manipulation language (DML) subschemas. Physical storage constraints are reflected through the schema to the subschema making the data structure different from the firm's data structure.

Olle points out that the user's needs--the (firm's) data structure--should be identified first, then the physical storage and access method (schemas, subschemas) can be considered. In the real world, the user sees a network data structure, in which a data entity may have more than one owner. For example, in a manufacturing parts list, many common parts go into more than one higher level assembly. Charles W. Bachman of G.E. points out that many types of structures that apparently are tree structures will not stand up under close scrutiny. Olle agrees, and adds that network structures are forced into tree structures because they are more understandable and familiar because they resemble structures of families and organizations.[13] Another reason for the popularity of hierarchies (trees) rather than networks may be man's limited ability to perceive the complexity of networks; hierarchies reduce complexity by combining entities at ever higher levels. People try to force network structures to fit tree structures, for ease of understanding, but in the back of their minds, they know that they are dealing with networks.[14]

The rest of this section on data structures examines sequential, random and list data structures, identified by Dodd as being building blocks basic for networks, trees, and other complex structures which are combinations of these three structures.[15]

Sequential organization. The sequential data organization is one of the most popular forms of data structure, due to the many applications which require data to be sequenced by some attribute, such as product number, employee number, or social security number. Also inherently sequential magnetic tape is still the cheapest way to store large volumes of information. In disk-oriented systems, where other structures would be more efficient, sequential structures are often used because some data processors fear the disorganization change and new methods might produce in a performing system. Thus, the sequential organization is the backbone of many of today's information systems because of storage volume costs or because of faulty data management procedures.

The attribute by which the sequential file is ordered is called the key, which may be, for example, an employee number, a department number or a social security number. Sequential lists also differ in the direction in which they may be traversed (see Figure 1). A one way list can be traversed in one direction only, equivalent to being able to read a tape only in one direction; a two way list can be traversed in either direction, equivalent to being able to



SEQUENTIAL STRUCTURES

Figure 1[16]

reach a magnetic tape either forward or backward; a circular one way list allows the search to begin at any point since the last recrod is connected to the first, equivalent to an automatic tape rewind capability.

Sequential structures provide fast access for a series of closely related records, as for example, a search of all people having names beginning with "N", if the file is sequenced alphabetically on the last name attribute. Sequential files stored on random access devices allow a binary search result for one person's name in n searches of a file of size 2^n or less. For example, "Newport" may be found in 7 searches of any file having less than or equal to 128 ($2^7 = 128$) names. In summary, sequential structures facilitate search of data record groups, but search for a single record is inefficient unless the file is stored on a random access device.

Disadvantages revolve around file update which often requires that the entire file be reorganized to insert a new data record or extend an existing record.

Random organization. The random organization overcomes the updating difficulties of the sequential organization by assigning random real addresses to the attribute key instead of sequential addresses. There are two feasible ways to obtain the address (1) dictionary look-up and (2) calculation. In dictionary look-up the direct address is calculated when the record is stored and its address is recorded in the dictionary. The dictionary itself is a sequential file, subject to the update difficulties previously mentioned. In addition, the dictionary must be accessed before the data can be obtained.

The calculation approach has its own unique advantage and problems. Often direct use of absolute physical address as the key avoids both look-up and computation (other than multiplication by unity) and is a special but important case. In other cases a mathematical algorithm (a hashing technique) is employed to generate an address from the attribute, an example being an arithmetic calculation based on the alphabetic positions of the first n letters in the attribute. Often a few combinations calculate to the same address, this causes collisions which must be circumvented by overflow pointers. The key of the record is always examined for a match, if no match occurs, the address in the overflow pointer is searched for a match, etc., until a match does occur.

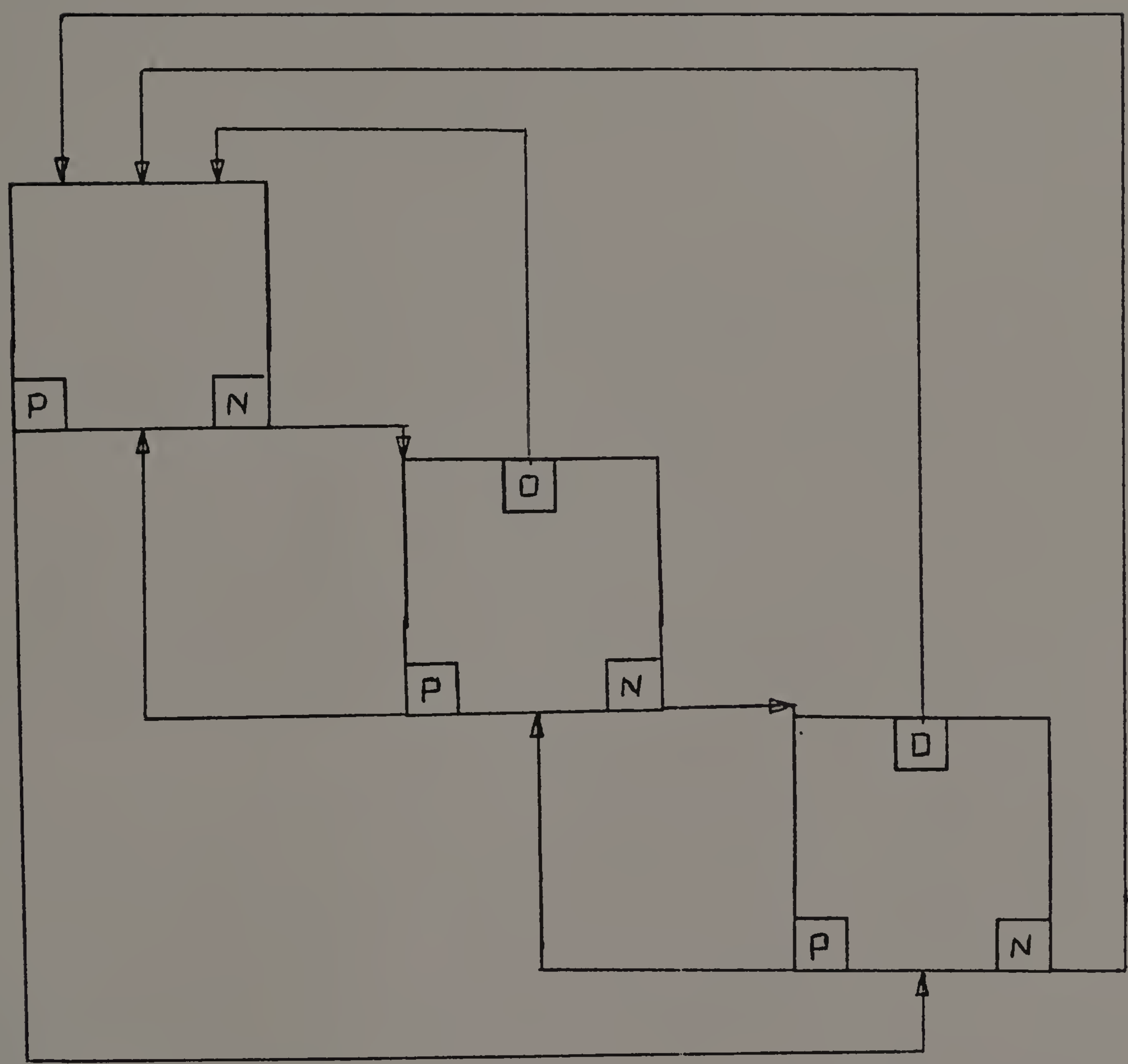
Random organization requires some direct access storage device (DASD). The dictionary look-up trades an easier update (the dictionary need only be resequenced) for an extra step in retrieval and the extra dictionary storage. The calculation technique allows any record to be retrieved in a single access; update requires no file or dictionary rewriting. Disadvantages of the calculation technique are that an extra overflow location is required in each record;

also, the technique cannot efficiently access a group of records due to the seek time lag of most direct access storage devices.

List organization. The use of overflow pointers suggests a structure which relates each record in turn by a pointer in the previous record; pointers may refer to any combination of the next record, the prior record, or the owner record (see Figure 2). The Codasyl Data Base Task Group refers to this structure as a chain.[17] List organizations may be classified as simple list, inverted list, or ring structure.

The simple list structure (chain with next pointer) has only next pointers and therefore can be traversed in only one direction, search must always begin at the first record. The ring structure on the other hand, has pointer wrap-around, the last record's pointer addresses the first record. It is also common for ring structures to have prior pointers to allow search in both directions and to allow easier update. The chain may be easily broken if both the prior and the next records are known. A more complex structure might have pointers to an owner record (see Figure 2).

Though this data structure provides a great deal of flexibility, the pointers may consume more storage than the data. Also in complex structures several next, prior, and owner records may pass through the same data record. In fact every data element may be an access key and have a



LIST STRUCTURE (CHAIN) WITH NEXT, PRIOR AND
OWNER POINTERS

Figure 2[18]

unique set of pointers.

Here again, the dictionary may consume more storage than the actual data. The organization is particularly suitable for situations where the data retrieval requirements are unpredictable, since all data is accessed with equal ease. Dictionary size may be controlled by inverting only one or two keys, the records themselves being organized sequentially or randomly. An immediate problem may be that the pointers may consume more storage than the data. Also in multipath structures, the deletion or addition of one data record or data item may result in substantial pointer maintenance.

When the pointers are stores in an array by category, matching operations may be performed on the pointers instead of the data. Such a structure is known as an inverted list, being that operations are performed on the pointers instead of the data. CODASYL refers to such a structure as a "Set declared as a Pointer Array".[19] As an example, if one wanted to find a programmer living in New York, the pointers of categories "New York" and "Programmer" would be processed for matches.

Complex structures. Complex data structures are constructed by combining simple structures into a more complex form to meet user response requirements in a hardware limited environment. Some examples of more complex files are multilist and cellular multilist with paged data. (Refer to

Appendix C, for an explanation of paging and virtual memory). A multilist file is a list structure having several multiple access keys per record. A cellular multilist file is a combination of multilist structure and inverted list.[20]

A tree structure is a hierarchical structure described in graph theory as having no circuits. Put differently, there is only one way of getting from any point Z to any point X. The structure looks like a tree with branches extending from a trunk (root) in all directions. A more complex structure is that of a network which may contain rings. Unlike a tree which may have but one owner record for one or more member records, a network may have several owner records for one member record. Tree structures portray a one-to-many relationship, network structures portray a many-to-many relationship. Many data relationships are in reality networks, but man has not the ability to understand the complexity of a network so hierarchical structures are assumed.

Physical Data Structure

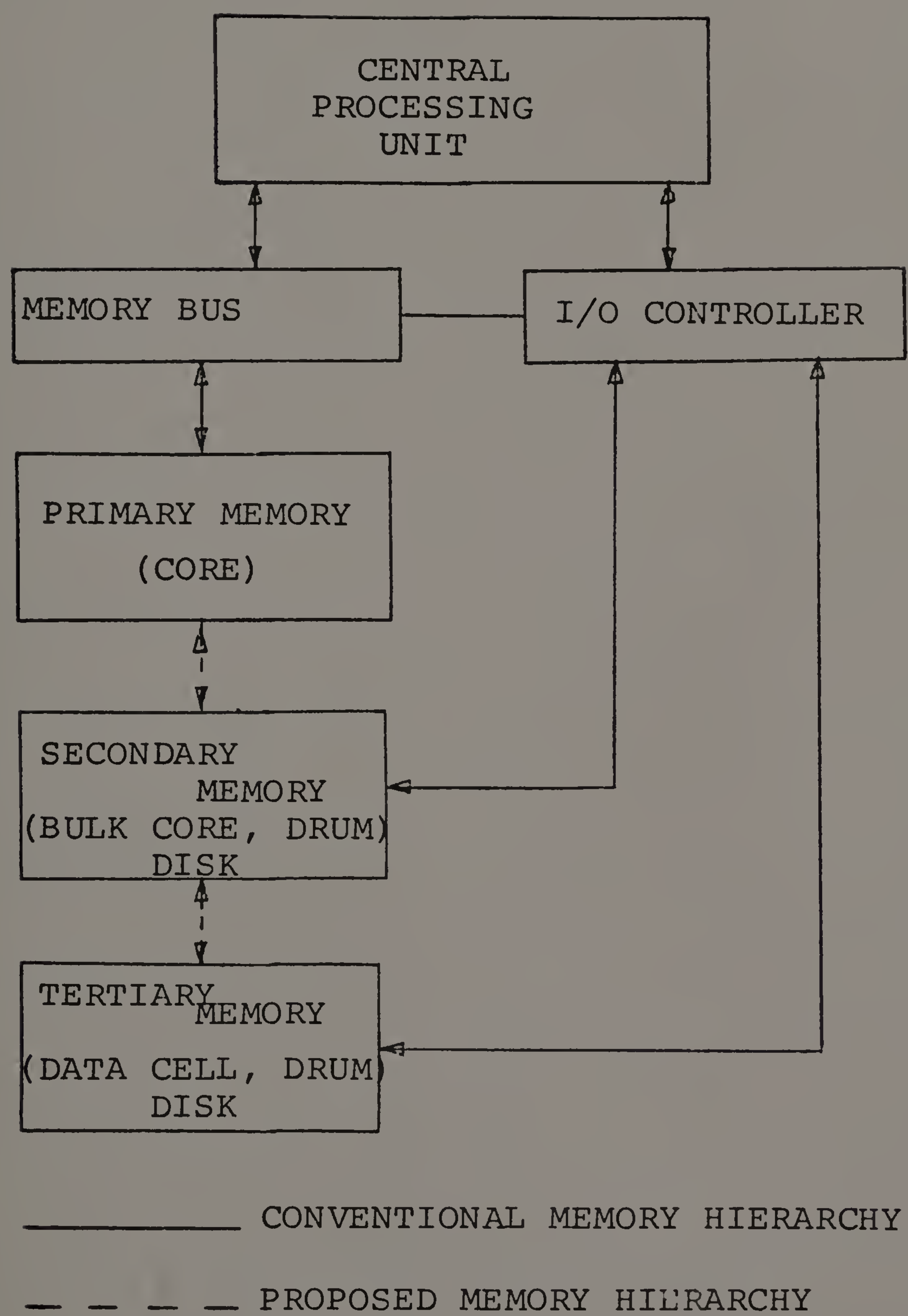
Design physical data base to meet operational-level requirements. Data base structure technology attempts to match physical structure capability to user data and response requirements. If physical structure could match the complexity of the firm's data structure, each user at every level could access needed data in sufficient time to control the firm's processes. This is not the case, and so, the data base physical structure is designed to match the operational level system, the firm's mainline information flow, as closely by category as possible. Then operational level response time, the most critical for management control of the firm's processes, will be minimized.

Hardware constrains physical data structure. The reason physical data base structure cannot match the firm's logical data structure lies in the discrete nature of hardware devices. Magnetic tape stores data linearly and magnetic disks and drums have finite length tracks and sectors. In addition their response is dependent upon mechanical movement. Main computer memory on the other hand is randomly accessible at electronic speeds. Two data items with widely separated locations may be accessed concurrently with identical response time.

Robin Williams comments that "if computers were built with associative memories, then random data organization would be used."[21] It is difficult to conceive of any structure but sequential in a magnetic tape environment, though in some cases network files have been handled by brute force methods with magnetic tape systems. Take for example, the parts list production explosion, numerous passes through the tape files with sort operations between passes are required.[22]

Magnetic disk and drum devices provide more flexibility than magnetic tape because data items and records are randomly accessible. The data structure can be designed to take advantage of the specific characteristics of moving arm disks; for example, records may be stored in groups on a single track. A pointer vector can then associate physical tracks with data contained in that track. Such an organization is called index sequential, because the records are stored sequentially and indexed by track. Usually 20% of track storage space is unassigned to provide area for file expansion. If a track becomes filled, a pointer to an empty track tells the access system where to continue the search; as many tracks become disjoint the mean access time increases and the data stored on the disk is reorganized by the Data Administrator. A major problem is the lack of a body of knowledge which will directly indicate that a portion of the data base is deteriorating either in efficiency or integrity and needs to be reorganized.[23] Chapter VI "Measuring and Analyzing Computer Systems and Data Base Effectiveness," presents a hardware/software monitor approach to the problem.

A hierarchy of memories. The fact that all data base elements do not have the same response requirements or are not accessed with the same frequency, coupled with the higher cost of faster responding storage hardware leads to the hierarchy of memories depicted in Figure 3. Primary memory is directly connected to the central processing unit through a memory bus. Lower memory levels are connected through an



A PROPOSED MEMORY HIERARCHY

Figure 3[29]

input/output controller. John Salasin proposes adding communications between the levels of memory.[24] In his proposal, the entire data base is stored on the lowest level DASD. As data is requested from the data base, blocks of data are transformed to higher level storage. The memory system might consist of fast semiconductor memory, core memory, extended core storage, disk memory and data cells. Salasin proceeds to derive equations to predict the effect of the proposed organization on data access times using sequential files, random access files and structured files employing multiple-hierarchical likked lists. Thus Salasin relates data structure to a proposed hardware hierarchy.

Dijkstra points out that hierarchies in computing hardware and software are a natural ocurrence, because of the huge gap between problem time, measured in hours or minutes and control processor time measured in nanoseconds (10^{-9} seconds).[25] A theoretical limit on information processing, $2x10^{47}$ bits per second per gram, has been derived by Bremmerman using two different methods -- quantum mechanics and thermodynamics.[26] Limits of computer switching speed appear to be in the neighborhood of 10^{-10} seconds, with 10^{-7} seconds for magnetic cores and 10^{-9} seconds for magnetic films.[27] The Josepheson switch will push switching speeds to the picosecond (10^{-12} seconds) level,[28] accenting the difference between machine time and problem time even more.

So, then the machine's best response time will eventually be in the picosecond range while data access need only be in the one second range for most manufacturing control system applications. Any data access requires a search through the computer's hardware and software hierarchies. The fastest computer memory is usually called a cache memory, it provides a small high access content addressable memory for recently accessed works.[30] The DAT (direct address translation) hardware in IBM's Virtual Storage System performs a similar function. (Virtual Storage and the DAT functioning is explained in Appendix C). Any reference to the main memory first interrogates the cache memory to see if the information is there, and only if not, is access made to the main memory. Cache memories are usually superfast TTL (transistor-transistor logic) semiconductor devices, slower (MOS metal oxide) semiconductor devices are usually used as mass storage devices or as extended core in the hierarchy.

A whole new raft of technology, including magnetic bubble memory, and their cousins, charge-coupled devices are emerging. Appendix D contains a description of the emerging memory technology. A characteristic of the technology is that its content is directly accessible by the CPU, just like core memory. An exciting development is the emerging optical memory. It is capable of replacing the entire hierarchy of core, drum, and disk systems now used, and thereby simplifying the whole architecture of computers.[31] The result is that there will be no peripherals whose access time must be masked, and therefore, multiprogramming and virtual storage will be eliminated, and data structures simplified.[32] For the present, however, the data base physical structure design is constrained by a hierarchy of discrete parameter memories.

Analysis of physical data structures. There is no established system of attack to the data base physical structure design problem. Four approaches to the problem have been to examine the literature for[33] (1) documentation of existing file structures used for a particular application, this documentation explains the advantages of the structures used over others that could have been used; or (2) examples of theoretical structures not directly associated with a particular application, these studies identify data structure requirements as parameters which are then used in comparison models to show the efficiency of one particular data structure over another, given specific requirements; (3) articles which examine a few structures which appear to be basic to more complex structures, and evaluate advantages and disadvantages of each of the basic structures, this gives the designer some basic modules with which to design the data structure, (4) articles which describe various structuring schemes as variations of a single structuring concept, this approach gives a continuity to the data structure. The most difficult part of the design problem is to predict the activity type and volume a particular data structure must meet after implementation. Data base content will grow in physical size and number of separate data items identified. Thus the data base physical structure problem becomes the reorganization problem.

The reorganization problem occurs for several reasons. Often it happens that a data base set which previously was frequently referenced no longer needs to be, and therefore, should not be maintained on a high speed storage device. It may be more cost-effective to move the data set to a less costly, lower speed storage medium. In another event it may be that the data structure no longer meets some access criteria and a restructuring is necessary. Another reason for restructuring may be that the data has outgrown the physical storage device and the integrity of the data is in jeopardy. Or a data-item addition or deletion of characters may cause a restructuring.

In the constant reorganization and input of new data to the data base it is likely that errors in accuracy and integrity will be introduced. It is impossible for a dynamic system to contain no errors, but these errors must be controlled. The quality of the data can be determined and improved only by working on the data base. Operations research techniques often can be used to estimate data distributions so that extraneous items may be flagged for closer scrutiny. Also, it might be possible to approximate a data distribution from other variables using regression. The result can then be compared with the true data distribution. Often correlation techniques will show various data-items to be highly correlated. If the correlation factor changes significantly perhaps it is due to integrity degradation of the data base.

Data Security in the Data Base

Control of access and manipulation. Once data is selected to be entered into the data base an access security profile is given to it. The Data Dictionary flags all data items and records having the same security level so that they can be stored in contiguous areas if possible. There are three commonly used methods of obtaining a secure environment in the data base. The method most often used is the password, used at the system level, application level, task level, or data record level to provide access to only specific individuals or functions.[34] Another method is physical location of terminals possessing special hardware identification devices. For example, a terminal located in the personnel department may have a special hardware answer-back which identifies it to the data management system as having access to only personnel data. Another method is the use of some physical aspect of the user, it may be a badge, card, or key or some personal characteristic such as the user's voice print. Currently there appears to be a boom in the market for badge readers, monitoring systems, and other forms of access controls.[35]

A system viewpoint. There are basically three areas in which data may be compromised, at the terminal, in the communication network, or within the CPU. There is currently a debate over whether computer installations should put their security systems within the main computer or some external monitor, which can oversee accesses to the main system and prevent unauthorized persons from compromising the system.[36] The reasoning behind an external security system is a clearer division of authority and understanding of the security system. A single person outside the main operating sphere controls the security system; operating people need not understand the system to perform their function, thus they must first understand the "black box" security system before they can compromise it.

Donn B. Parker of Stanford Research Center, identifies the methods of penetration of systems as computer facility accesses, software trapdoors, masquerading as a legitimate user, piggybacking, password detection and wire tapping. These are the means, he says, the threats are the potential and actual actions of people. The nature of people-produced threats include circumstances of people's actions, their ability to act, the procedures they use, and the technical methods they can employ. It is most important, therefore, to take the position of looking outward through the external interface of the system to the surrounding environment to successfully discover and analyze threats.[37]

Dr. Ware and Professor Glaser, two security experts, indicated to R. G. Canning that most present systems are so complex that no one person can comprehend them, this leaves security gaps which can be used by intent violators.[38] Professor Glaser is a skilled system penetrator. Working as a security consultant it is not unusual for him to penetrate a "secure" system within five minutes time at a terminal, after learning standard system operating procedures for the system and thinking about the matter for a time.[39]

Baran of the Rand Corporation states an additional security principle that at first seems surprising. The principle is: if one cannot safely describe a proposed security system in the unclassified literature, then it is not sufficiently secure to be used with confidence.[40] A system gains its logical power by standing up under scrutiny, preferably by a single individual who can see all the ramifications. A system gains its protective power by providing such a large number of possible combinations that finding <u>the</u> combination is very difficult.

<u>Present state of security</u>. The security market is in its infancy, encrypting routines are offered but not widely used, even in announcing a major IBM security project, Chairman T. V. Learson admitted the market was underdeveloped and customers were not clamoring for new developments or pushing

the manufacturers for new devices or systems. Currently, IBM is the only computer manufacturer with a large security project. Everyone is waiting for tehhniques to be developed by government and private research groups, to preclude spending a great deal of research money on possibly unproductive projects.[41]

Another factor contributing to the current market morass is that present computer-based information systems do not contain a great deal of company sensitive material. In addition, sophisticated security enhancements in a system which does not really contain extremely sensitive data, only adds increased complexity to an emerging data base management system.

Security in data base management systems. Most data base management systems provide the types of security routines necessary to control security in today's systems. An example is the DBMS-imbedded routines which report access violations to the Data Administrator. The subschema itself is a type of security device. In a subschema the data names may be different from those in the schema, hence outside individuals cannot discover even the names the authorized user has for his own data.

Other security considerations. One aspect not discussed refers to checkpointing and backup and recovery methods to protect the system during accidental shut downs. This technology is discussed in the January, 1972, EDP Analyzer.[42]

Nor was mention made of the effect power fluctuations may have on data reliability.[43] Transformers and power stabilizing devices are available to protect against power fluctuations.

Data Management Systems

Development versus purchase or lease. A firm wishing to implement a data management system (DBMS) has two choices, it can either purchase and modify an existing system or develop its own special purpose system. Developing a special purpose DBMS requires functional statements of user need and specifications of capabilities to be developed, as well as a development plan, development monitoring and finally testing. This approach is only useful for large organizations.[44] Custom programming of the data base management functions constitutes a large initial programming cost and very large maintenance cost.[45]

Therefore, the direction of the data management field has been toward use of commercial generalized data management systems. As the cost of computer processing has come down, users have been more willing to accept the inefficiencies of generalized data management systems in return for reduced programming costs.[46] Fred Withington, at the 1971 ACM National Conference, predicted that users might "cheerfully (accept) software overhead exceeding 90% as the price willingly paid for a degree of adaptability, flexibility, and human orientation that would be incredible today."[47] The use of a generalized DBMS is one step in this direction.

Generalized data management system capability. One advantage of using a generalized DBMS is that is is continually evolving. Canning uses the term Data Management System (DMS) to describe systems having broader functions than merely data base management. Often a DMS is initially marketed with only the core data base requirements of a limited data definition capability, a limited number of access methods, and the public data files, other capabilities will be added later.[48] A list of possible DMS functions gleaned from the CODASYL Feature Analysis of Generalized Data Base Management Systems and the Mitre Data Management Systems Survey follows:[49]

- Data Base Management system
    - Data definition language
    - Data manipulation language
    - Modular routines at record level
    - Subsetting function
    - Optimal storage structure-access method combinations
    - Common storage structure-access method combinations
    - Common access to multiple data files
    - Standard interfaces with other system software
- Communications control program
    - Interrupt analysis, input/output control
    - Resource allocation, terminal and line service
    - Code translation
    - Application program support
- Retrieval Subsystem
    - Search routines, employing complex conditional expressions
    - Extract routines
- Processing subsystem
    - Data validation
    - Update (at data-item level)
    - Customized procedures capability
    - Arithmetic capability

Output subsystem
- Sort-merge
- Report preparation
- Display preparation

Other elements
- Data security function
- Restart and recovery function
- Data compress/decompress function
- Data encode/decode function
- Query function

From the preceding list of capabilities which either exist in data management systems or are planned, it is obvious that the security system, data maintenance routine and generalized file structuring methods already exist in the DBMS and need not be designed by the Data Administrator or computer staff. The problem becomes one of selecting the proper DBMS from the myriad possibilities existing today.

Data management system selection. T. W. Olle writing in 1970 guesses that conservatively there are about 10 to 15 host language systems commercially available, and probably as many as 100 self-contained systems.[50] A host language system is an enhancement to a procedural language such as COBOL or PL/1, Fortran, Basic or APL, examples of the systems are IBM's IMS or GE's IDS. Enhancements are oriented to transfer of data between levels of memory hierarchy, and the user must be proficient in programming with the host language. Self-contained systems provide a unique language, supposedly suitable for the parametric user as well as the programmer. An example would be the Codasyl Data Base Task Group proposal[51] or IBM's GIS or Informatics Mark IV.

The Codasyl Systems Committee finds the difference between the two systems hard to delineate and predicts that the difference between the two approaches will disappear as data management systems continue to evolve.[52]

The selection of a DBMS requires the development of a specification and an evaluation technique.[53] A good place to begin is to study the Codasyl Feature Analysis and the Mitre DBMS Survey reports which examine typical systems. Today, no DBMS has anywhere near all the features enumerated in the above list. Canning suggests the supplier be examined to ascertain that the system will continue to evolve, and be compatible with future hardware upgrades or conversion.[54]

Tom Gilb, a consultant in Oslo, Norway thinks that the DBMS should be evaluated along the dimensions of portability, implementability, maintainability, operational cost, software reliability, supportability, and data security and reliability.[55] Specifically, he says that the vendors do not know enough about the above factors in absolute terms, and know less in competitive terms. He argues for a simple self-developed package based on standard Cobol/Fortran routines. His most piercing comment is the one concerning software error rates. Stating that no one knows the theory behind software bug control, he notes that IBM's OS is out of control. Bug rates have increased from 1000 bugs per release in 1969 to 11,000 bugs per release according to IBM's own 1972 bug report indexes.

Users fear being locked into a DBMS as they were once locked into hardware because of hardware dependent software. It is generally true that DBMS's lack the degrees of flexibility they will eventually possess. Canning notes that he has been reporting on the development of data management systems in EDP Analyzer for 6-1/2 years,[56] he concludes that data management systems are really in their infancy, and there are many tough problems still unresolved.[57]

Irrespective of the facts, the generalized DBMS is increasing in popularity as evidenced by the many special workshops currently being held.[58] Also the Codasyl Systems Committee plans to publish a report in the Fall of 1973 on DBMS selection and implementation.

The selection of the DBMS is dependent on the requirements of the particular firm intending to use the system. The implementation bottleneck lies in reducing data redundancies existing in independent systems. The implementation procedure is that portrayed in Appendix A, implement the mainline information flow, a module at a time; expand the system to higher levels.

Data Base Administration Only a Part of the Data Function

This dissertation is concerned with the management of data. In yesterday's batch-oriented systems Data Management and Data Base Administration were almost synonymous. Data enters the system at the computing center, is validated by the main computer and enters a data base; later it is accessed by a batch-oriented report generator or application. In these systems data is under the complete control of the Data Administrator. But computer-based information systems are evolving into a system of computers and remote applications, the telecommunications system connecting the various parts. Data no longer exists only within the computing center but in remote input areas. It seems logical therefore, that the data function should extend into these areas also. Data may be rendered unreliable or compromised at the input device or within the telecommunications system just as easily as within the computing center.

Therefore, data management must include data responsibility within the data input system and the telecommunications system. The data function name should be Data Administration, indicating a breadth lacking in Data Base Administration. The rest of this dissertation examines the elements of the function in these systems and the technology understanding requisite of the new Data Administration function.

FOOTNOTES

1. R.L. Laska, "All The News That's Fit to Retrieve," Computer Decisions (August, 1972), p. 18.

2. Ibid.

3. R.G. Canning, "Creating the Corporate Data Base," EDP Analyzer, Vol. 8, No. 2 (February, 1970), p. 12.

4. Ibid., p. 3.

5. W.W. Cooley and P.R. Lohnes, Multivariate Data Analysis (New York: John Wiley & Sons, inc.), p. 170.

6. R.G. Canning, "Processing the Corporate Data Base," EDP Analyzer, Vol. 8, No. 4 (April, 1970), p. 12.

7. Ibid., p. 13.

8. S.S. Ruth and P.J. Kreutzer, "Data Compression for Large Business Files," Datamation, Vol. 18, No. 9 (September, 1972), pp. 62-66.

9. Canning, "Processing the Data Base," Op. Cit., p. 12.

10. CODASYL Data Base Task Group, April 1971 Report, (New York: Association for Computing Machinery), pp. 13-14.

11. "The Data Catalogue," Synergetics Corporation, One Garfield Circle, Burlington, Massachusetts 01803.

12. T.W. Olle, ed., "The Large Data Base, Its Organization and User Interface," Data Base, Vol. 1, No. 3 (Fall, 1969).

13. R.G. Canning, "Organizing the Corporate Data Base, EDP Analyzer, Vol. 8, No. 3 (March, 1970) p. 6.

14. R.G. Canning, "Trends in Data Management, Part I," EDP Analyzer, Vol. 9, No. 5 (May, 1971), p. 9.

15. G.G. Dodd, "Elements of Data Management Systems," Computing Surveys, Vol. 1, No. 2 (June, 1969), pp. 119-127.

16. CODASYL Data Base Task Group, April 1971 Report, (New York: Association for Computing Machinery, p. 26.

17. CODASYL Data Base Task Group, April 1971 Report, Op. Cit.

18. Ibid. p. 50.

19. Ibid., p. 51.

20. Dodd, Op. Cit., p. 128.

21. R. Williams, "A Survey of Data Structures for Computer Graphics Systems," Computing Surveys, Vol. 3, No. 1 (March, 1971), p. 4.

22. R.G. Canning, "Organizing the Corporate Data Base," EDP Analyzer, Vol. 8, No. 3, (March, 1970), p. 7.

23. J.K. Lyon, "The Role of the Data Base Administrator," Data Base, Vol. 3, No. 4 (Winter, 1971), p. 12.

24. J. Saldsin, "Hierarchical Storage in Information Retrieval," ACM Communications, Vol. 6, No. 5, (May, 1973), pp. 291-295.

25. E.W. Dijkstra, "The Turing Lecture," ACM National Convention, Boston, Massachusetts (August 14, 1972).

26. W.H. Ware, "The Ultimate Computer," IEEE Spectrum (March, 1972), p. 88.

27. Ibid., p. 89.

28. "Josephson Switch: Faster than Transistor," IEEE Spectrum, Vol. 10, No. 4 (April, 1973), p. 87.

29. Salasin, Op. Cit., p. 292.

30. C.G. Bell and A. Newell, "Computer Structure--Past, Present, and Future," Fall Joint Computer Conference, 1971, p. 389.

31. "RCA Uses Halographic Approach," IBM Puts 8K on a Chip," Computerworld (February 28, 1973), p. 43.

32. J.G. Roberts, "Asymmetric Memory Hierarchies," ACM Communications,Vol. 16, No. 4 (April, 1973), pp. 214-215.

33. L.D. Martin, "A Model for File Structure Determination for Large On-Line Data Files," International Federation of Information Processing Society, Occasional Publication No. 3, File Organization Selected Papers from File--1968. An I.A.G. Conference, Swets and Zeitlinger N.Y. Amsterdam, 1969.

34. M. Weinstein, "Who Accesses What on Remote Terminal?," Computerworld (December 6, 1972), p. 24.

35. "Security Device Market, Still in Infancy," Computerworld (December 6, 1972), p. 46.

36. E.D. Lundell, Jr., "More Secure (But Vulnerable) Systems Expected Soon," Computerworld (December 13, 1972), p. 13.

37. E.D. Lundell, Jr., "Inflexible DP Systems Said to Attract Dishonesty," Computerworld (November 1, 1972), p. 2.

38. R.G. Canning, "Data Security in the CDB," EDP Analyzer, Vol. 8, No. 5, (May, 1970), p. 5.

39. Ibid., p. 2.

40. P. Baran, On Distributed Communications: IX Security, Secrecy, and Tamper-Free Considerations, The Rand Corporation, RM-3765-PR, August, 1963, p. 7.

41. "Security Device Market Still in Infancy," Computerworld (December 6, 1972), p. 47, see also D. Leavitt, "Encrypting Routines Offered But Not Widely Used," Computerworld (December 6, 1972), p. 18.

42. R.G. Canning, "Computer Security: Backup and Recovery Methods," EDP Analyzer, Vol. 10, No. 1 (January, 1972). p.

43. M. Weinstein, "Power Fluctuations Can Cause Users More EDP Problems than Brownouts," Computerworld (November 8, 1972), p. 16.

44. E. Raichelson, "Two Methods for Data Management Systems Acquisitions," ACM SIGFIDET Work Session: "The Information System and Data Base Management," New York City, March 29, 1973.

45. R.G. Canning, "Trends in Data Management, Part 2," EDP Analyzer, Vol. 9, No. 6 (June, 1971), p. 6.

46. Ibid.

47. A. Drattel, "Eniac to 2001: "Economic Payoff" and 90% Overhead?," Computerworld (August 11, 1971), p. 2.

48. Ibid., p. 7.

49. Codasyl Systems Committee, *Feature Analysis of Generalized Data Base Management Systems* (New York: Association for Computing Machinery), May, 1971, and Mitre Corporation, *Data Base Management Systems Survey* (Springfield, Va., 22151: National Technical Information Service), AD 684707, May, 1969.

50. T.W. Olle, "MIS: Data Bases," *Datamation* (November 15, 1970), p. 50.

51. *Codasyl Data Base Task Group, April 1971 Report*, *Op. Cit.*

52. Codasyl Systems Committee, *Feature Analysis of Generalized Data Base Management Systems*, (New York: Association for Computing Machinery),May 1971.

53. Raichelson, *Op. Cit.*

54. R.G. Canning, "Trends in Data Management, Part 2," *EDP Analyzer*, Vol. 9, No. 6 (June, 1971), p. 9.

55. T. Gilb, "Generalized Data Base Packages Not Worth Cost/Risk," *Computerworld* (May 16, 1973), p. 12.

56. R.G. Canning, "Trends in Data Management, Part 2," *EDP Analyzer*, Vol. 9, No. 6 (June, 1971), p. 6.

57. *Ibid.*, p. 12,

58. SIGFIDET and SIGBDP are sponsoring a special two day program at the 1973 National Computer Conference, New York City (June 7-8, 1973), entitled "Managing the Impact of Generalized Data Bases." SIGFIDET has also sponsored a workshop held in New York City (March 29, 1973), entitled "The Information System and Data Base Management," which was filled to overflowing.

CHAPTER III

DATA INPUT SYSTEMS

Introduction

Data input system critical to data base accuracy and integrity. The data base input system is crucial to the functioning of the data base. Data accuracy, integrity and security responsibility must begin at the input system. The typical Input, Processing and Output functions have long been used to characterize the data processing operation. The model remains valid, even for data base centered information processing operations. The processing function may be tailored to an end user's needs and then further refined or changed as user needs change. In a like manner the output reports may be reformated, etc., until the system meets user needs. The input function, however, is critical. It is performed once, the data becomes part of the data base and its accuracy and integrity is difficult to improve. Therefore, if the Data Administrator is to be responsible for the accuracy, integrity and security of data, this responsibility must begin with data input system design responsibility.

Accuracy checks remote from sensing. In the past, data input, complete with data accuracy and integrity checks, were always implemented at the main processing computer. Data was keyed onto cards from source documents, and the keying verified at a keypunch facility. Then the main computer checked the data using error routines to perform limit checks, type checks and other system checks necessary to eliminate extraneous data from processing. Cards having data lying outside the checks were returned to the keypunch operation and possibly the originating unit for correction. Processing delays for such data could be significant, and worse, the originating unit may not remember what the correct input should have been! As data base management system technology advances provide the environment for growth in number and variety of computer-based MIS applications, the effect of incomplete, inaccurate, untimely data is heightened and may become critical for operational-level systems.

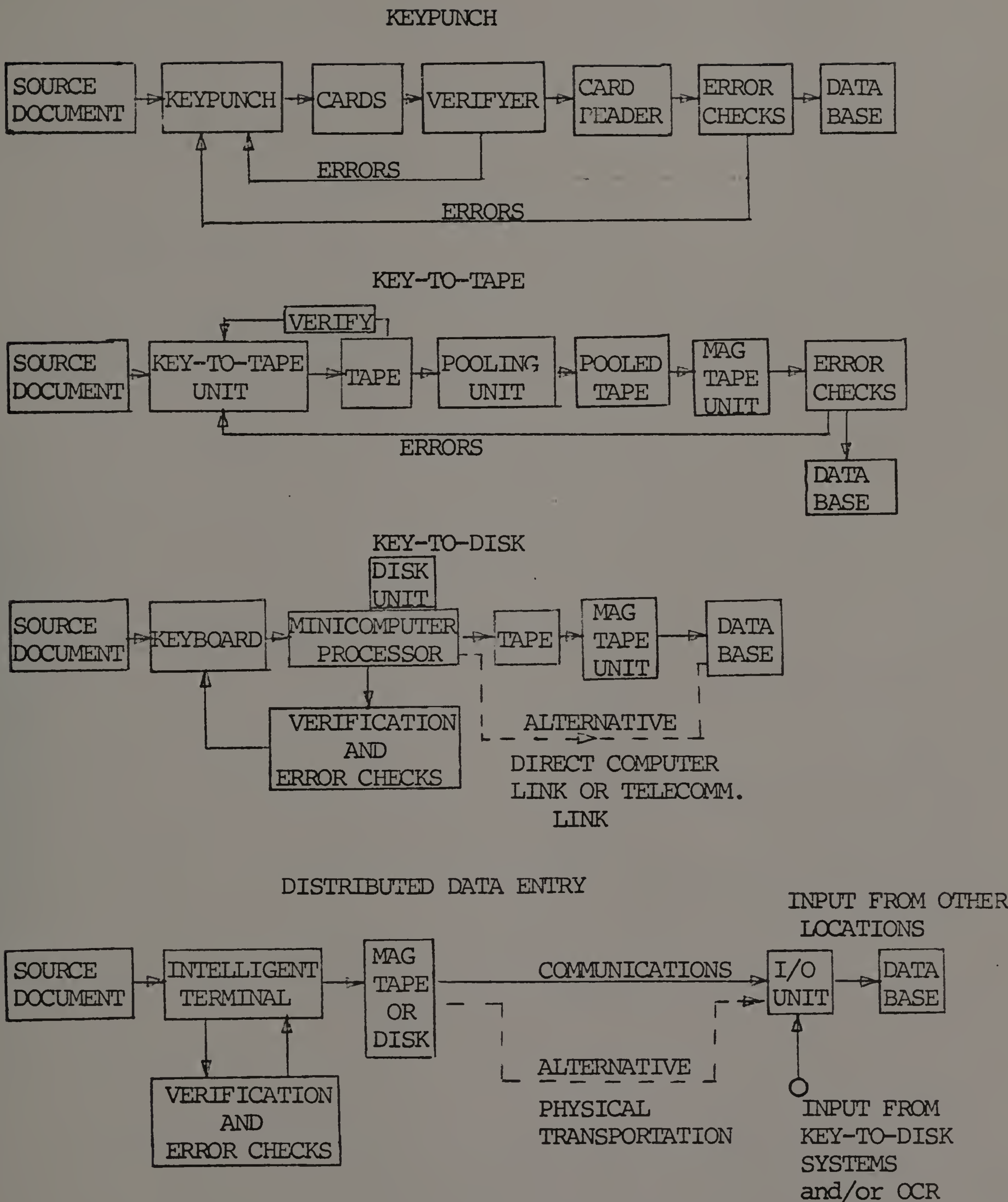
Accuracy checks move closer to sensor. A solution to the data quandary is to move the accuracy checks closer in time and place to the data sensor. The data originating units is where data input accuracy responsibility should lie. The person who collects the data is the individual most likely able to decipher error messages and correct data input before it enters very far into the information system.

Howard Sunderland of Western Union said he has done several studies on error rates and finds that 90% of the errors occur after data has left the data source. Dick Baker of Chemagro notes that if user departments enter data, often they can flag incorrect reports before entering them.[1] The data may be corrected at a later time or at least be denoted as "iffy" within the data base.

The technology allowing the user to have data responsibility is a conglomeration of minicomputer-based key-to-disk systems and intelligent terminals. These systems provide for on-line accuracy checks during the keying operation. Most checks once done by the central computer can now be performed at the input station.

Figure 1 portrays the data entry system evolution from centralized error checking intelligence to distributed intelligence data entry. The first step in the evolution is keypunch-based data entry systems having separate punch and keystroke verify operations and logical error detection at the central computer. The stand-alone key-to-tape machine moves the verify operation into the data keying operation as an on-line feedback process. Minicomputer-based key-to-disk systems incorporate the verify operation and the error analysis operation, thereby providing immediate correction capability. The final step in the evolution is to provide a mixture of intelligent terminal, and key-to-disk systems and other input media in a distributed intelligence manner, so that all data entry is performed as close to the the data generating source as feasible.

<u>D.B.A. accuracy and integrity responsibility begins with the data input system</u>. The importance of competent data checking at the input source is crucial to the accuracy



DATA ENTRY SYSTEM EVOLUTION

Figure 1[2]

and integrity of the data base. Once invalid data enters the data base it may over time trigger other data or generate invalid relationships, data and relationships which must later be corrected. Late data entry causes similar problems. Because timely linkages may have not been generated, the data to be linked may now be nonexistent or recently restructured.

It is for these reasons that the Data Administrator's data integrity and accuracy responsibility must begin at the data input system. Effective data responsibility must encompass the design and implementation of the data input system. Data security, though not mentioned in this section, also begins with the data input system. Part of the security system may lie within the remote input/output device.

The analysis which follows examines the elements of data input systems and then develops an input system design philosophy which will help assure total system data accuracy and integrity for any environment.

Card Punch Systems

Mainstay of data input systems. The card punch is still the mainstay of the data input system. An International Data Corporation Research Study indicates that the card punch with an installed base of 250,000 units and a 29,000 unit per annum growth rate continues to be the most important data input device. Keypunch replacements, however, are gaining with a 10,800 unit per annum growth rate on a 58,600 unit base. Keypunch growth is leveling off while keypunch replacement growth accelerates and threatens to displace card punch importance.[3]

Reasons for dwindling popularity. Several reasons for the movement from cards to other systems obtain, notwithstanding the cybernetic advantage attendant in moving the data base input function closer to the organization's sensing unit. The shear cost of cards in a high volume environment to say nothing of the cost of card handling, storage and disposal can be significant in the decision to move to key-to-tape or key-to-disk systems. Other reasons center on increased management control of the data input operation. A key-to-disk system is minicomputer based and collects operator workload and performance statistics. Roy R. Lovelace of the National Education Association points out that such data can be used as a basis to single out individuals for recognition and praise where under other systems their performance would be lost in an aggregate of statistics.[4]

Key-To-Tape Systems

Key-to-tape systems do not gather statistics. Mostly, they are stand alone units which put data onto a minireel of tape instead of cards. They have several advantages. Because input is keyed directly onto tape, records are not limited to 80 characters. Also, program cards do not have to be changed for different jobs as program control functions reside within the terminal. Another advantage is that built-in keying verification capabilities confirm input without transfer to a separate machine. (This advantage is attendant also on the buffered card punch.)

Key-to-tape, however, has the disadvantage that mini-reels are difficult to handle. And if several operators work on the same job, their reels must be combined in a separate pooling operation. Advantages are that cards and the verification procedure is eliminated.

Key-To-Disk Systems

Data checks at keystations. Key-to-disk systems evolved from key-to-tape systems. In key-to-disk systems a minicomputer controls and monitors input from several keystations. All the features of key-to-tape systems which eliminate cards and the keystroke verification process are included, but in addition the power of the minicomputer is used to perform check digit calculations, range checks, control totals and format validity. These are functions previously performed on the central processing computer. Often utilities to perform these checks require 40Kbytes within the main computer, but require only 4Kbytes in the minicomputer controller due to the simpler interactive environment and specialization of routines to jobs.[5]

<u>Computer staff loses data accuracy responsibility</u>. This new power of key-to-disk systems is changing the way data operations are performed. Whereas once data production was the responsibility of the keypunch operation and data validity and editing were the responsibility of the computer staff, the data entry operation now has responsibility for production, validity and editing. Data accuracy and integrity responsibility has moved closer to the data source. The computer staff has no data verification responsibility and their old validating routines, based on 80 column records are useless, since the format has changed.

What has been occurring in many companies is that the 80 column format is retained and dual checks are run on data at the keying operation and again preceding processing. Further, some managers are recommending the 80 character record be standardized so that they may return to the card punch operation should key-to-disk fail. This is the age-old problem, people are hesitant about becoming dependent upon operations they cannot control.[6] The answer I see is to centralize data system responsibility in the Data Administrator Function.

<u>Centralization of data authority optimizes</u>. Centralization of data authority permits system optimization. Key-to-disk systems may be linked to the central computing systems, eliminating the current disk to system input tape translation. Full key-to-disk advantages could be realized

as errors are caught and corrected as they occur in keying, not at the computing center. The central computer is freed of the administrative tasks of data entry. And finally, increased throughput can be realized through automatic validation, and flexibility is achieved by the ability to key data to suit the form of the source document, not the 80 column card.

Operator statistics--uses. In addition the statistics generated lead to better management of the input operation. This is the first case in which the work of every individual is monitored for volume and accuracy in an industrial setting. But statistics have another use not immediately apparent to the worker, they can be used to identify poorly designed source documents. And new designs may be studied for immediate analysis.

The operator monitoring is disregarded by the operators themselves, who feel a sense of pride in working with key-to-tape and key-to-disk systems, according to Ann Conte, New England training director for Mohawk Data Systems.[7]

Key-to-disk savings. The reason companies are moving to key-to-disk systems is that savings are so great, often one application alone can justify keypunch replacement. Everet Lawton of New England Electric, said that his company saved $1,500 per month on machine rentals alone on the customer accounting application.[8] Jim Feeley of Philco-Ford ran an experiment with key-to-disk and key punches. He selected six applications and divided input between the two systems. The key-to-disk system generated output 45% faster than the keypunch system. Another analyst found that eight operators working on key-to-disk systems were equivalent to 13 operators working on key punches.[9]

Paul Sidikman in describing the Reader's Digest implementation said that a 10% productivity increase was expected at each of 142 centrally located keystations driving three key-to-disk systems. He found a 25% increase on easy jobs, such as change of address, where the machine takes over many simple tasks. No improvement was found on complex tasks such as those in which a subscriber letters instructions in the margin of a letter. The operator spends little of her time actually keying.[10]

Environment for key-to-disk. The key-to-disk implementation requires that all documents be at some central location, since keying stations must be within a few thousand feet of the shared processor. And data volume must be sufficient for at least five key-stations to make this approach feasible. If these requirements cannot be met, another solution could be the use of intelligent terminals or on-line data entry.

Intelligent Terminal Based Systems

Intelligent vs. special purpose terminals. Intelligent terminals have some ability to be programmed to meet the needs of a variety of specialized applications.[11] Often, the flexibility of an intelligent terminal is not needed, in this event the logic can be "hardwired" at reduced cost. The variety of Point-of-Sale terminals now being actively marketed is one example of the hardwired approach. This cost effective approach may be only an interim solution, Caxton C. Foster predicts that by 1980 the computer element of an intelligent terminal possessing the power of an IBM 7090 will cost less than one dollar.[12] If this is true, the special purpose terminal will disappear.

If we accept the definition of "intelligent" as being adaptive to a variety of specialized user needs, then the class of intelligent terminals also includes small business computers and accounting machines which have the ability to communicate with other machines. The flexibility of intelligent terminals is evinced by their ability to emulate special purpose terminals and often to communicate with a variety of computers by simply changing their communication software. They may also operate in a stand-alone basis, but this is not necessary to the definition.

The basic purpose of an intelligent terminal is (1) to put some of the processing at remote locations, (2) to provide more effective data entry techniques and (3) to compress data for faster and more economical data transmission.

Intelligent terminals ideal at remote sites. Intelligent terminals provide for processing of input data at locations remote from the central computer. Louis Morton of Kaufman and Broad, for example, describes how they are used in his company.[13] Kaufman and Broad has many remote offices with small non-computer-oriented staffs. Previously, they had used punch tape and the mail system to collect data at a central computer. Tapes received from remote locations had to be spliced, sorted and error rates were high since punch-tape is not easily readable by humans and punch-tape terminals give no error messages.

The application of intelligent terminals facilitated matters. They were easy to program so that a clerk could call the desired input program. The operator then had only to fill in the form on the screen. Input errors were immediately noted and diagnostics generated by the terminal's software. Error correction was effected at the input source where the operator, being closer to the data than someone in the computing facility, could more accurately determine what the correct input should be.

Martin McDonough of Fanny Farmer Candy describes how a network of Sycor terminals handles order entry in the process connecting the plant, distribution centers and corporate headquarters. The whole new operation he says costs less than the keypunch-printing-mailing system it replaced. The best feature of the intelligent terminal is the flexibility of on-line, off-line use. An additional plus, again, is the "tremendous" editing and formatting capability possible through programming and the use of cassettes. The conventional editing to include, zero fill, field justification, data type checking and very rapid check digit calculations are all handled in less than 4Kbytes. Comparable generalized routines would consume 40Kbytes on the main computer.[14]

Edward Neff and Craig Jester of American Trading and Production emphasize the increased accuracy of data entry with intelligent terminals because they place the data entry function into the hands of the people responsible for the source document, and thus reduce the length of the communication channel. They saw a network of intelligent terminals as the only way to solve the error problem without the large CPU and data transmission cost of an on-line data entry system.[15]

The old procedure required punching orders and shipping data from source documents at three sites. Error utilities implemented at the central computing facility would flag errors and return bad data cards to the originator. The process of original data preparation, error detection, return to originator, and return to central processing averaged two months! The intelligent terminal network reduced the time for order entry and invoicing to 3 days and inventory is now up-to-date within 2 days. Furthermore, the error rate is cut by more than 90%. This is due to order entry clerks now being responsible for data preparation and data entry. This is possible because the terminal allows the data to be entered in familiar source document form, the terminal logic formatting data correctly for entry to the IBM System/3 Model 10. Keypunch operators are eliminated at an annual savings of $500,000.

Intelligent terminal evaluation factors. Martin McDonough of Fanny Farmer gives an indication of the dimensions on which intelligent terminals should be evaluated. The cassette capability is probably the most important. It was mentioned by all discussants throughout the 1973 Computerworld Computer Caravan. One should also consider the programming language available. Dr. Dick Simons of Texas A & M thinks that "the only thing blocking wider use of programmable terminals is a very definite software gap." A gap being currently closed. Sycor, for example, announced a Terminal Application Language in the summer of 1972.[16]

Other evaluative factors concern the kind of usage expected and the type of printer required. The type and size of CRT screen are also considerations. Two screens may display an equivalent number of characters but the matrix sizes may differ. For example, a 1200 character display may consist of 15 eighty character lines or 20 sixty character lines. Special features available, such as automatic tabbing from one field to the next are also important. Neff and Jester mention level of maintenance support provided by the terminal manufacturer, expandability of the system and unattended send/receive along with stand alone capability.

Special purpose preprogrammed terminals. Special purpose, preprogrammed terminals such as point-of-sale (POS) operate like intelligent terminals but are specially designed to cope with just one or a few, well-defined jobs very efficiently. They are used on the sales floor, at the shipping dock, in the warehouse or in the business' back office. The goal of these systems is to collect more information, faster and more accurately while keeping the clerk's effort to a minimum. On the sales floor POS terminals record sales and serve to update inventory. The shipping dock use is to record transfer of merchandise to truckers, customers or vendor returns. The back office uses other terminals to generate merchandise tags, payroll information, input purchase orders or accept invoice information.

The console coaches the user through the transaction. Edit routines check responses and use is made of light pens, mark sense devices, magnetic stripe-card readers and other elements designed to collect data without depending upon the accuracy of the operator. The terminal may then generate a printed English product description which is clearer to the customer than present cash drawer output.

Hahn's Shoe Division of U.S. Shoe uses POS terminals to avoid expansion of the headquarters' keypunch operation.

The division has been expanding at a rate of about ten retail stores per year. Speed and accuracy are important to the system users because salesmen are paid by commission. Each store is polled at night and each salesman's account is credited and the inventory is adjusted. Errors are reduced by one half over the key-to-tape central operation and reports are generated much more rapidly. Inventory reports are now completed in 3-5 days instead of 10 days. "Hot" items are recognized sooner and inventory may be balanced between stores to match market demand.[17]

Teachable terminals. Another type of terminal termed "teachable" is being developed by Threshold Technology Incorporated. The teacher portion is housed in a cassette. In voice input systems the terminal is capable of recognizing 100 words with 99% accuracy. The cassette contains the voice pattern of the user who has repeated each word ten times to catch varying voice inflections. It takes ten seconds for the system to learn the user's voice. During operation, if a match is not found a "beep" is sounded and the terminal halts until recognizable input is received. This system adds efficiency to the grocery check-out procedure because the buyer hears the prices and the checker has hands free to bag purchases.

On-Line Data Entry Systems

On-line data entry not for everyone. Richard Allen, director of MIS at Con Arga, Incorporated, calls intelligent terminals an "evolutionary" step between the keypunch and direct entry.[18] Sid Smith of Crown Zellerbach says the on-line interactive criteria is based upon (1) the value of the information and (2) the effect of time on value. In other words, is the value of moving the information quickly worth the cost to the organization served? An example he gives puts a lumber yard on one side of a river and a paper production facility on the other. The old method was to send a bill of lading on the barge with the lumber. Clerks processed the paperwork, and then organized the unloading operation. The on-line system sends information across the river as soon as barge loading is completed. Clerks on the other side of the river, documents in hand, are on the dock awaiting the barge.[19]

Another application at the First National Bank of Boston is not so clear cut. There on-line direct entry is used to cope with large volumes of high value input records, which therefore, must be handled carefully. The service charge for each transaction is small, but it was thought that the greater editing features of an on-line system were necessitated by the cost of error. A side benefit was a 25% staff reduction.[20]

The advantage of on-line data entry sought here was the ability to clean out errors as they occur, rather than save them for a later edit cycle. Accounts Receivable, stock transfers, freight payments and mutual funds are handled by this system. Twenty-five thousand transactions are handled during an eight hour day. Data base update, however, is reserved for the evening hours, because data base update time constraints are not crucial but data security is vital.

Herb Cronin of the First National Bank of Boston makes an interesting comment. He says that once equipment is selected for a real-time system--"the die is case."[21]

Applications determine the data input system. The purpose of on-line data entry is to eventually eliminate the data center and put data entry into the user department. Terminals for data entry may then have dual use as data base inquiry devices. Robert Cheris of Lumber Mutual Life sees cassette-based terminals, intelligent terminals and POS terminals as providing better efficiency by correcting errors as they occur in using departments.[22] In contrast Paul Sidikman of Reader's Digest is interested in throughput which can be best attained by centralizing data input operations. He says that using departments have typists or clerks input data, many of whom are incompetent keying operators.[23] Reader's Digest applications are well-defined and keying efficiency is the central issue since highly unusual circumstances do not often occur.

The central point in the design of any input system is that applications define the system, not hardware. Further, many organizations do not need on-line systems. Immediate data base update is not critical and on-line data base query is unnecessary. Daily or weekly update may be sufficient in which case an intelligent terminal and a switched communication system will meet the user's needs. For example, only five minutes is required on the average to transmit all the data generated in one day at a Hahn Shoe Store. The Fanny Farmer system has similar needs. Thus unattended polling at night is sufficient to meet data needs. The only difficulty with unattended operation is that a janitor may turn off the terminal. Advantages of these systems over paper tape systems is the editing logic inherent in the terminal.

Many on-line systems do not have direct access to data base change but instead construct an intermediate disk file. The data base is then updated at a later time. This eliminates needless restructuring and the use of many overflow files as well as insuring data base security and integrity. Only a system as critical as an on-line reservation system need immediate data base update.

On-line CRT systems reduce paper handling. Another change brought about by on-line systems and the increasing use of CRT terminals is the elimination of paper reports. Dick Baker of Chemagro says that the effect of CRT terminals in his company is that reports are no longer received and purchase orders are held only temporarily. Data is always available at the CRT if required.[24] A recent survey by Northern Nekoosa Corporation found CRT output growing at

86% to 45% for hard-copy. If hard copies were necessary from CRT's, in most of the cases the user writes down the information and then uses an office copier or the user records the information, his secretary types it and then uses a copier. Only in 15% of the cases did the user address some hard copy output device like a line printer and only in one case was a hard-copy CRT attachment used. But 71% of the respondents hoped to have CRT's with hard-copy output within five years.[25]

Computer graphics. CRT growth is the result of new technology which reduced CRT cost from $15,000 to $8,000 by eliminating the refreshing requirement. A disadvantage of these storage tube devices is that they cannot be used for graphic input since light pens are ineffective. Terminals with graphic capabilities must, therefore, be augmented by some low cost graphic tablet. One such device uses an ordinary piece of glass to conduct acoustic waves along the X and Y coordinates. A finger or pen interrupts the waves much as the Doppler effect interrupts sound waves and the points are input to the computer.[26] Many of these graphics terminals are interfaced to low cost minicomputer controllers. Computer graphics appears to be catching on as something more than a toy. They are being ordered to do specific jobs, with reasonable certainty that they will prove effective in those tasks.[27]

Machine Readable Input Systems

Optical character recognition systems. Machine readable input devices include Magnetic Ink Character Recognition (MICR) and Optical Character Recognition (OCR) systems. OCR includes character readers dominated by IBM and Control Data, bar code readers dominated by Addressograph-Multigraph and mark-sense readers dominated by IBM and Optical Scanning. Presently, there are approximately 1500 to 2000 installed character readers in the U.S. representing a 200 to 300 million dollar market.[28]

Most OCR devices recognize one of two fonts, OCR-A or OCR-B. OCR-A is highly stylized, with very angular capital letters, numerics and special characters popular in the U.S. OCR-B has both upper and lower case letters, numbers and special characters. It is easier on humans but harder on machines. Other OCR devices learn new fonts as the need arises. For example, Scan Data's 250/350 uses a sheet of paper with samples of the desired font to learn the font. Limits of acceptability are controlled by software. Any font can be learned including Japanese Katakani. The system scans documents, storing read characters on a disk from which unrecognized characters can be extracted and corrected before collected on magnetic tape.[29] Some OCR devices can recognize hand printing, such as those used to recognize zip codes in large, automated postal distribution points. They often cost millions of dollars and are not cost-effective for most business applications.

Economics of OCR. To many users the cost of OCR may be prohibitive and the capabilities so incredible that users may be scared away. TWA has replaced OCR units with a key-to-disk system. It was stressed that OCR needs a high volume and suitable applications to be effective. A reject rate of just over 1% could eat up savings of $160,000 per year at TWA. The cost savings were the result of lower cost input devices, supplies and operator salaries. IBM selectrics with OCR fonts could be used and proofreaders checked 85% of the documents. Typists and proofreaders are more abundant than keypunch operators, hence the training required and wages are less. Inherent advantages also obtain because the format flexibility of an OCR document is almost infinite allowing correction of any part without disrupting the entire document. But OCR devices are very sensitive to humidity. TWA has used the system effectively for five years.[30]

Zayre Corporation has just implemented an OCR system in the Accounts Payable section. Previously a logjam had built up in the data entry area due to physical and personnel constraints. The environment of volume with required quick-turnaround was evident. Zayre found OCR to be sensitive to handprinting quality and turnaround printed documents. An operator was subsequently assigned to check the printer output for voids or unreadable fills.

OCR system design. Noel Goulston, the Methods Manager at Zayre gave some design pointers. The form should be designed so as to keep the reference mark out of the way of the person's hand when writing to avoid smudging. Check digits are rarely misread if preprinted in gothic font. Make full use of editing for protection such as check digits, hash totals, field verification, maximum values, customer number. And mylar ribbons are preferable to nylon. Zayre realized savings of $4,600 per month on this one application (12 keypunches were replaced) and a speed increase of 80 to 120 times that of keypunch/verify.[31]

OCR evaluative factors. The advantage of OCR is its greater accuracy, the result of moving the data input interface closer to the data source. The closer the interface moves to the data source, the more accurate that data becomes, and the more displacement occurs of human converters.

One reason given for the slow acceptance of OCR other than a lack of the right volume/speed constraints is that the use of OCR means substantial changes in procedures and therefore high implementation and training costs. The change impetus must remain the increasing cost of the labor content of data entry.[32]

J. Robert Taylor, Marketing V.P. at Inforex sees a mixture of OCR and key entry emerging. But the market he says is highly applications dependent, aimed at large users of turn-around documents such as banks and credit companies.

The final evaluative statement is best made by Arthur W. Holt in closing a Datamation article on OCR Terminals. His comments are applicable to any data input system evaluation. He says, "it is very well for an author to say that OCR is wonderful, that prices are too high, etc., but it is more useful to add some specifics to these nebulous benefits that OCR is supposed to confer. Unfortunately, most of these answers cannot be obtained by engineers or philosophers. Most of the answers must come from the users of OCR systems."[33] In short we do not know enough about the man-machine interface and its cost effective solutions to make it a hard science. We are still at the descriptive stage.

Mixed Media Systems

Redundant input. There is currently a trend in data input systems to provide for redundant input to protect against failure of the prime input stream. This is especially true in heavily time dependent, production-oriented workload environments. For example, OCR can read MICR or hand printing, and key-to-tape systems, key-to-disk systems and some on-line terminal systems can be used interchangeably with little system modification, and hard copy input forms such as invoices and order forms may include keypunch columns. There is always danger that data input source documents will be mutilated or the input system hardware will fail. The concept of mixed systems provide alternate routes for getting data into the system within the time span required to maintain control of the firm's operational system.

Source of data expertise. The various data sensing locations throughout the organization must be considered in totality by one person or organizational function. The Data Administrator works with users to determine their present and future data requirements, his job is to know the sources of data generation within and external to the organization. His knowledge of each data source must be sufficient to be able to ascertain the accuracy of the data entered at that source. Data accuracy at the input source is crucial to data base accuracy as is knowledge of the data path reliability to the data base system. (Data path knowledge will be discussed in the next section.) If the Data Administrator is to have data accuracy and integrity responsibility he must control the data input system and the telecommunications function.

Disseminate data responsibility but centralize data control. Data input control includes specification of primary and secondary (redundant) data input sensors and the means to be used to determine data accuracy at the sensor. The data sensing department is most likely to be able to specify checks necessary to ascertain data reliability within some system specified limit. The D.B.A. works with the data sensing department to ascertain that the proper data input equipment is used to meet total system efficiency requirements and sufficient data checks are performed to meet system data reliability specifications.

The system may take the form of a mixture of keypunches, key-to-tape devices, key-to-disk systems, OCR devices and intelligent and preprogrammed terminals operating on-line or off-line depending upon data value and time constraints.

The basic idea running through a philosophy of data input system design is to move the data input function from the data center to the input sensor which may be a data center itself, if data volume is sufficiently high as in the Reader's Digest subscription example. As Howard Miller of the First National Bank of Atlanta put it, "There are a lot of smart cookies out there" and letting them do more improves the overall efficiency and quality of the work being done. If an operator is responsible for the entire job, the operator could edit incoming documents and catch errors before they are keystroked, reducing the error rate. Included is user department participation in programming error detection routines particular to their department.[34]

Remote data bases. J. Robert Taylor, marketing vice president at Inforex, sees a future in putting operational data bases at their point of most frequent use. Remote data bases are made possible by the minicomputer which allows processing at the point of data use. Grace Hopper sees the future as systems of computers.[35] And figures contained in the latest government inventory of computers confirm that there is a trend toward minicomputer use for many jobs.[36] The general movement of the corporate data base to distributed locations increases information system speed and reliability at the operational level and gives added weight to the argument that Data Administrator responsibility must also be distributed to match the geographical dispersion of the data base and the input system.

The next chapter will discuss the data communications system which provides the means for a geographically distributed data base and input system. The data accuracy and reliability increase resulting from moving the input system closer to the sensor may be negated by a poorly designed and planned data communications system.

FOOTNOTES

1. "User's Ride 'Intelligent' Bandwagon," Computerworld (April 25, 1973), p. 11.

2. Adapted from F.H. Reagan, Jr., "Is the Keypunch Really Obsolete?," Computer Decisions (April, 1973), p. 14.

3. "Keypunch Replacement Shipments Expected to Increase This Year," Computerworld (September 29, 1972), p. S14.

4. "What's Real Story On Replacements?," Computerworld (September 27, 1972), p. S1.

5. D. Leavitt, "Users Told to Evaluate Cassette Language Features," Computerworld (February 21, 1973), p. 5.

6. M. Weinstein, "Key-Disk? Discard Safe Approach," Computerworld (April 11, 1973), p. 4.

7. "How to Get Those Operators," Computerworld (September 27, 1972), p. S11.

8. D. Leavitt, "One Application May Justify Keypunch Replacement," Computerworld (February 21, 1973), p. 4.

9. M. Weinstein, "Key-to-Disk Best for Central Use," Computerworld (April 4, 1973), p. 4.

10. "Reader's Digest Multifaceted Approach Has 142 Stations for Key-to-Disk Units," Computerworld (September 27, 1972), p. S2.

11. J.N. Bairstow, "The Terminal That Thinks for Itself," Computer Decisions (January, 1973), p. 10.

12. C.C. Foster, "A View of Computer Architecture," Communications of the Association for Computing Machinery, Vol. 15, No. 7 (July, 1972), p. 558.

13. M. Weinstein,"Key-To-DiskBest for Central Uses," Computerworld (April 4, 1973), p. 4.

14. D. Leavitt, "Users Told to Evaluate Cassette Language Features," Computerworld (February 21, 1973), p. 5.

15. "Accuracy is Intelligent Terminal's Chief Asset," Computerworld (February 28, 1973), p. 4.

16. "Software Gap May Be Retarding Use of Programmable Terminals," Computerworld (September 27, 1972), p. S20.

17. E.J. Bride, "POS Terminals Boost Data Accuracy, Cut Report Time," Computerworld (August 9, 1972), p. 3.

18. "Software Gap May Be Retarding Use of Programmable Terminals," Computerworld (September 27, 1972), p. S20.

19. "How Important The Data? How Much Speed?," Computerworld (April 11, 1973), p. 4.

20. D. Leavitt, "One Application May Justify Keypunch Replacement," Computerworld (February 21, 1973), p. 4.

21. R.A. Frank, "Advantage of Real-Time Systems: Clean Up Errors as They Occur," Computerworld (February 21, 1973), p. 7.

22. "Software Gap May Be Retarding Use of Programmable Terminals," Computerworld (September 27, 1972), p. S20.

23. "Reader's Digest Multifacet Approach," Computerworld (September 27, 1972), p. S2.

24. M. Upton, "With On-Line Data Entry, Users Work to Keep System Up," Computerworld (April 18, 1973), p. 4.

25. "Growth in CRT Output Use Booming," Computerworld (September 27, 1972), p. 4.

26. "Unit Puts Data Entry at Fingertips," Computerworld (August 16, 1972), p. 22.

27. "Better Hardware Spurs Graphics Terminal Rise," Computerworld (September 27, 1972), p. S22.

28. "OCR Market Expected to Grow," Computerworld (November 29, 1972), p. 52.

29. "Between Intelligent and Dumb Terminals There Lies 'Teachable' Units," Computerworld (September 27, 1973), p. S22.

30. "TWA System Allows Continuous Processing," Computerworld (April 18, 1973), p. 4.

31. "OCR Devices Mean Gravy," Computerworld (February 21, 1973), p. 4.

32. "OCR Market Expected to Grow," Computerworld (November 29, 1972), p. 52.

33. A.W. Holt, "Smart Terminals," Datamation (October 15, 1970), p. 57.

34. "Letting Operators Do More Improves Overall Efficiency," Computerworld (March 21, 1973), p. 4.

35. E.J. Bride, "Brainy Operations Staffs Are an Asset: McFarlan," Computerworld (July 5, 1972), p. 4.

36. See Computerworld: "Mixed Media" (April 4, 1973), "IBM Continues to Lose Grasp" (April 25, 1973).

CHAPTER IV

DATA COMMUNICATIONS

Introduction

This chapter and the one following consider data communications. Chapter IV forms a theoretical basis for understanding the relevant parameters in data communications design and Chapter V discusses the actual design and implementation of such systems.

Why use data communications?. George Glaser of McKinsey and Company says there are three main reasons computer users should use data communications instead of decentralized computers, (1) to provide access to a common data base, (2) to respond to remote events and (3) to provide economies of scale.[1]

Access to a common data base is typified by airline, hotel and automobile rental reservation systems in which data is extremely volatile, operational response time is short and input occurs at remote locations. Crime control, credit and medical information systems also require rapid response from remote locations.

Response to remote events is typified by tactical response to geographical market conditions by a large consumer-oriented business. For example, one store of a chain may have a run on a specific color, size or style of garment and need to readjust inventory so as to maximize turnover and reduce the number of lost sales.

Economies of scale and increased operational system reliability is one more factor moving corporations toward data communications. Numerous remote users are often able to share a more powerful hardware-software complex of specialized machines at less cost than if they possessed independent computers generalized to a variety of tasks. The Tymshare system and the Arpa network are two examples of this approach.

So, we see that the communications portion of the information system is becoming more and more important. Unfortunately, only a few data processing people have even a superficial idea of what happens in the communications network. And most common carrier personnel do not have any notion of how data is used in the corporate information system or how errors may affect data integrity.[2]

<u>Data Administrator role in data communications</u>. Data Administrator responsibility begins at the input sensor and ends with the data user. Along the path from input to user, data accuracy, security and integrity may be violated. The Data Administrator must have sufficient knowledge of the communications system to intelligently examine it, specify weak points and then have a sufficient degree of operational control over the communication system design and operation to secure data from harm. Then he may state the data accuracy, security and integrity levels of the various data elements to the user with full confidence in his assessments. Furthermore, in order to meet operational level response

requirements a proper match must occur between data input system, data output system, the structure of the corporate data base (whether it be centralized or distributed) and the linkage between these systems--the communications system. This may be likened to impedance matching between electrical and physical systems such that power transfer from one system to another is maximized and loss is minimized. The Data Administrator is the system engineer who controls impedance matching among subsystems to maximize data flow power at the least cost, this maximization results from a proper balance between data accuracy, security, integrity and system response.

To understand the important link between data source, data usage and data base content, the Data Administrator must understand how data is transmitted through the communications network as well as how data is transmitted and stored in the corporate computer. A knowledge of the present carrier system and transmission techniques is required to understand system response problems and point to areas in which data may become multilated, i.e. have its integrity destroyed or be rendered insecure. Special emphasis on usage of error detection and correction is necessary, since this service is not a part of the common carrier service and must be separately implemented. Finally, the response, security, integrity and accuracy of data is tied directly to how the data communications system is implemented at the corporate computer. User controlled data switching and processing functions may be either centralized within the computer or modularized and distributed. It is concluded that the distributed intelligence system for data communications provides the maximum degree of system flexibility and data reliability. Finally, the role of the Data Administrator in telecommunications is summarized in light of the foregoing analysis.

Historical Development of a Data Communications Market

Early history. Digital data communication over electrical wire was first demonstrated by Samuel F.B. Morse in 1844. A dot-dash direct current signal was transmitted between Baltimore, Maryland and Washington, D.C. In 1858 the first transatlantic telegraph cable was laid. Since these first communications were used to transmit written messages, Alexander Graham Bell's 1876 human voice transmission over wires was a breakthrough and led to the design of our present voice oriented systems. Intercontinental communication growth was slow. The first commercial transatlantic radiotelephony was implemented in 1923 but consisted of only a single voice channel. It was not until 1956 with the opening of a high quality 36 channel cable that transatlantic voice communication became popular.[3]

Satellite systems beginning with Telstar in 1956 and Early Bird in 1962 changed the complexion of international communications. Telstar was not a stationary or synchronous satellite which meant that antennae had to track satellite movement. This is equivalent to sighting a dime at fifty feet with a 380 ton gun (an accuracy of four minutes of arc).[4] Newer satellites are synchronous, that is, they revolve about the earth so as to coincide with its rotation and thereby appear to be stationary. Today, there are over eight such satellites providing 25,000 voice channels with growth to 250,000 voice channels predicted by 1982. The reason for this sudden growth of satellite communication is that technology--rocketry, LSI satellite circuitry and powerful traveling-wave-tube amplifiers for ground stations combined in a cost effective manner to provide desired communication service at less expense with greater reliability than either cable or over-the-horizon radio.

The present plant--adaptable to data communications?. The key to data communication use within present corporations must, however, begin not with satellite communication but with the use of domestic communication systems. These systems are the base on which corporations must build a data communications system. Many complaints are voiced against the communication services for their outdated equipment and lethargic growth, but R.G. Canning concludes after an analysis which extends over three months of EDP Analyzer that the data transmission service currently being offered by common carriers is adequate for data communications, if the users take certain actions.[5] These actions will be discussed in depth in the rest of this chapter.

Incentives of communications industry--sufficient for growth?. Looking back at the Bell Telephone Network Plant, we see that it is gradually being expanded in a manner that will turn it into a digital network. Already, one-half the total plant is five years old or less.[6]

LaBlanc and Himsworth look at the capital demand of the communications industry. Their data show that the communications industry required capital was 7 billion dollars (20% of corporate demand) during the period 1961 through 1965, 13 billion dollars (14% of corporate demand) during the period 1966 through 1970 and 10 billion dollars (18% of corporate demand) during the 1970 through 1972 two year period.[7] Clearly, the communications industry is expanding rapidly as indicated by the above figures. However, the monopoly is constrained by a fixed return-on-investment. Return maximization requires plant to be built up and old plant to be depreciated over a long period. A 40 year replacement cycle is used. Maintenance earns no reward. The result is that there is no incentive to increase plant usage, since increased usage will only lead to decreased prices to the consumer and no rewards for the common carrier.

LaBlanc and Himsworth conclude that the communications industry will have to make itself more attractive than its competitors in the capital markets in order to attract this large share of capital. Therefore, communications in this decade will be shaped more decisively by financial constraints than by any other factor.[8] There is a need to change the incentive structure of the industry to make it more adaptable to the fast changing communications technology.

A data communications market. The communications market growth and strength has and will continue to reside in voice traffic. As a typical example, General Foods Corporation detailed planning for data communications beginning in 1968. They found that over 95% of their communications traffic was voice and that by 1975, data communications would still represent less than 10% of their total load.[9] Yet, the general prediction is that the number of data transacitons will increase eightfold by 1980 and perhaps twentyfold by 1985.[10]

Isavk, writing in the Honeywell Computer Journal states that the 1965 prediction was that in 1970, 58% of all general purpose computers would use data communication; whereas, in fact only 13% utilized data communication in 1970. He cites reasons being that third generation equipment cutover was more difficult than anticipated and this slowed implementation of advanced applications. In addition, the market was not as large as previously believed, it is only the larger and medium sized companies which have sophisticated needs and sufficiently sophisticated capability to permit data communication implementation. The number of computers owned by these sophisticated users is small compared to the number of installed computers.[11]

We may then conclude that the data communications market is only a small portion of the total communications market. Its growth rate, however, is outpacing the growth rate of communications in general. Moreover, the present voice network will be the basic building block on which data communications will grow. The following sections will discuss the technical aspects of data communication.

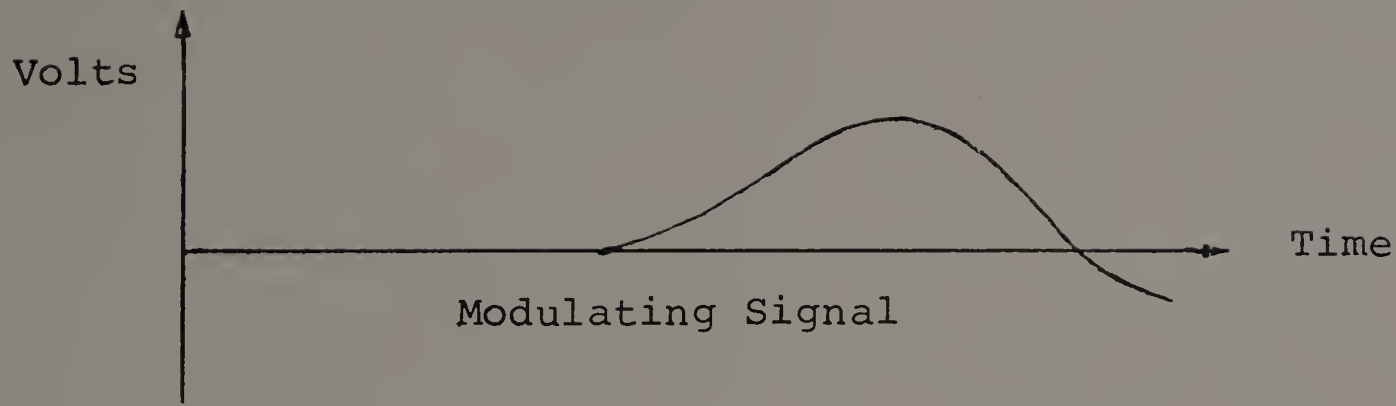
Data Transmission

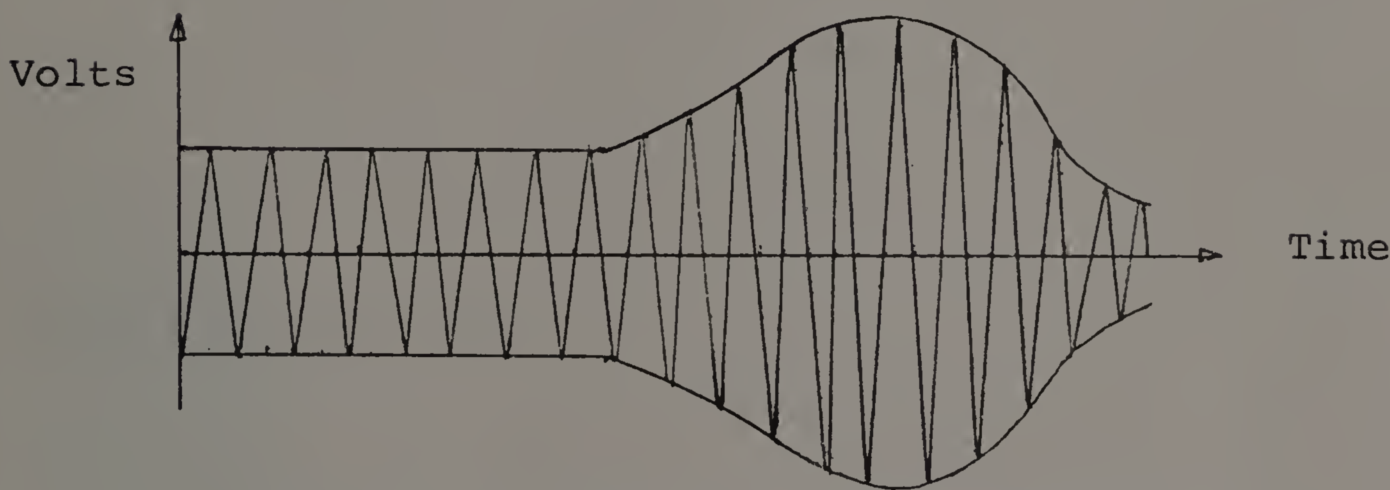
Transmitting the data signal. Modulation techniques. Transmission of information from point A to point B over an electrical wire requires that somehow the information be contained in the signal. The process of modifying the signal so that it contains information is called modulation. Information may be added to the signal by varying signal amplitude, fequency, phase or by switching the signal on and off as specified by some code.

Varying signal amplitude is called amplitude (AM) modulation. The signal usually is depicted as a varying frequency envelope, frequency being required to transmit the signal over a radio circuit (see Figure 1). In such a transmission a high frequency carrier frequency is added to the information content frequency envelope. The result is two separate frequency envelopes differing in phase, called carrier frequency sidebands. Identical information is contained in each sideband, therefore, transmission frequency spectrum may be saved by filtering out one of the sidebands and only transmitting one sideband. This is called single sideband (SSB) transmission. At the distant end the filtered sideband derived from the transmitted sideband is added back and the signal demodulated to the original amplitude varying low frequency envelope.

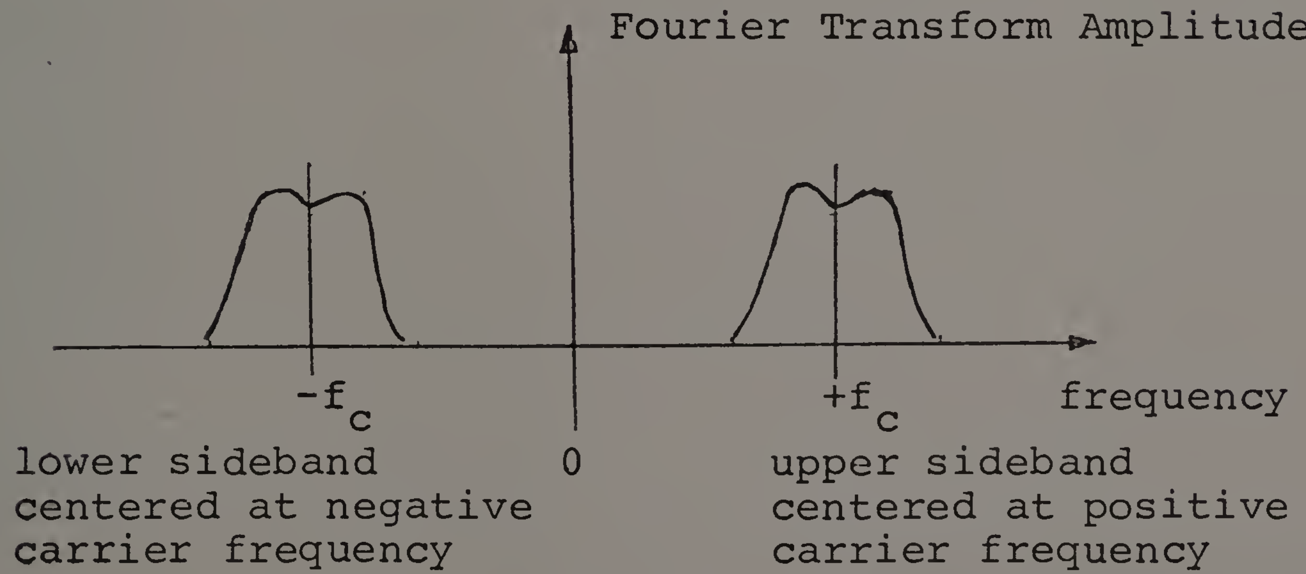
An alternate method of adding information to the signal is by modulating carrier frequency. This technique is called frequency modulation (FM). A disadvantage of frequency modulation is that it requires a bandwidth several times that required of amplitude modulation. But it is also an important advantage because no information is contained in the frequency's amplitude as is true with amplitude modulation. This becomes especially important when the signal is susceptible to extraneous voltage variations and random noise or static. Demodulators sense only frequency variations and voltage limiters can be used to protect demodulation circuitry from saturation by a voltage impulse. Engineers usually say that FM trades bandwidth for reduced noise. Frequency modulation is used in most microwave, satellite and tropospheric scatter communications systems.[13]

Phase modulation (PM) is similar to frequency modulation except that the information is contained in the phase





Amplitude Modulated Carrier Frequency



Relative Importance of Frequencies in
Amplitude Modulated Signal

AMPLITUDE MODULATION

Figure 1[12]

of the carrier signal rather than frequency of the signal as is the case with frequency modulation. In effect there is no essential difference between frequency modulation and phase modulation and the term, frequency modulation, is used to describe both.[14]

Pulse code modulation (PCM) digitizes the information signal and is the modulation technique most resistant to noise. The transmitted signal is a sequence of constant amplitude pulses arranged in binary-code groups. The immediate advantage of this technique is that the trains of pulses can be regenerated almost perfectly by any number of repeaters over any distance because the information content is not related to the amplitude of the pulse. However, as Canning points out, PCM systems must operate in synchronism with the distant end, and noise impulses may effect this synchronism.[15] Current communications systems are moving in the PCM direction.

Multiplexers and concentraters. Introduction. Multiplexers take modulated signals and mix them for more efficient transmission. Multiplexers combine several low speed channels for transmission over a single high speed channel. A characteristic of multiplexers is the bandwidth on the output side is equal to the sum of the bandwidths on the input side of the multiplexer. Line concentraters on the other hand combine a number of low speed circuits into a higher speed circuit where the bandwidth of the high speed circuit is <u>less</u> than the sum of the low speed circuit bandwidths, sometimes this causes <u>contention</u> among the input signals for the output channel. Concentraters then differ from multiplexers in that they use traffic statistics to save bandwidth.

Bandwidth and channel capacity. Bandwidth is a measure of the information carrying capacity of a communication channel. In digital circuits for example, a 10 bit (binary) per second data rate will require a frequency bandwidth of 10 cycles per second. (Hertz, after the noted electronic experimenter is now the accepted terminology for cycles per second.) The rule for digital transmission is that one hertz is required to transmit one bit of information.

In general, there are two system limits on the amount of information per unit time a channel can transmit. The first limitation is based upon the inherent line capacitance and inductance which has a tendency to store information, and therefore, impede instantaneous response to signal changes. The second limitation is the inability of the system to distinguish transmitted signal from noise. These limitations can be tied together in a simple expression for system capacity:

$$C = \frac{1}{t} \log_2 n \quad \text{bits/sec}$$

where 1/t is the bandwidth. In the expression C is channel capacity, t is the minimum time required for a system to respond to signal changes and n is the number of distinguishable signal levels.[16] For digital networks n is two and system capacity is equivalent to bandwidth. This equation and the rule of a minimum bandwidth of one hertz for one bit of information can be transcended by some elaborate combinations of modulation techniques.[17]

Multiplexers. Two commonly used multiplexing techniques are frequency division multiplexing (FDM) and time division multiplexing (TDM). In frequency division multiplexing the signals are arranged side-by-side to constitute in effect one wideband signal. Advantage is taken of the fact that speech signals have a bandwidth of four kilohertz whereas twisted wire channels are bandwidth limited some 25 times higher. Twisted wire generally connects users to the local central office.

Time division multiplexing samples various input signals repetitively and their instantaneous amplitudes are transmitted sequentially. Bell system implementation has switches at each end of the line rotating synchronously across twenty-four contacts. The sampling rate is once every 125 microseconds by switches rotating at 8,000 revolutions per second.[18] More sophisticated TDM techniques are minicomputer based.

Concentraters. The time-assignment speech interpolation (TASI) method used for high cost, long distance communication networks is an example of the use of traffic statistics to form a concentrater. Canning notes that concentraters use traffic statistics to save bandwidth, but a drawback of such systems is the contention of low speed lines for paths within the high speed circuit.[19]

Time-assignment speech interpolation capitalizes on the brief silences in normal conversation to achieve channel savings. When a speaker pauses, the channel he held is reassigned. Using this technique 96 overseas circuits can carry 235 simultaneous conversations. Traffic statistics indicate that a speaker actually talks less than 40% of the time he is connected. TASI monitors each speaker's line and compares its signal 25,000 times per second against five distinct reference levels. The speaker is assigned a channel only when he is judged to be active. The speaker retains a channel for the second or so taken by a word or a group of syllables. Only after a pause of one fourth of a second is the channel taken away from the speaker. When the speaker resumes talking, he is likely to be on a different channel.[20]

Rudin complained in a 1970 IEEE Spectrum article that traffic theory, the study of the statistical flow of messages is largely responsible for economical voice transmission as it exists today, but is only beginning to be applied to data-communications.[21] He sees a great use for a multiplexer-concentrater with buffer storage which would switch and queue messages and data, thus eliminating the mismatch between low speed terminals and high speed channels. All computerized message switching systems and computer-based telecommunications systems to be discussed at length later are in effect multiplexer-concentraters.

Modes of transmission. Synchronous versus asynchronous. There are two modes of transmission, synchronous and asynchronous. Synchronous data communication systems transmit a continuous stream of data, the send end must be synchronized in time with the receive end, hence the name. Asynchronous communication systems are start-stop in nature, send/receive commands being transmitted with the data.

Advantages are inherent in both systems. The asynchronous system wastes bits because a start bit and a stop bit must be added to each character. The synchronous system on the other hand must transmit blank characters even when no data needs to be transmitted. Thus synchronous systems lend themselves to efficient transmission of high data volume while asynchronous systems are better when it is necessary to communicate with a wide variety of terminals operating at widely different data rates.

Several methods obtain for achieving synchronization. One method requires sending signals until sync is established, synchronization may then be maintained in one of several ways. A sync signal could be transmitted at regular intervals in a portion of the frequency spectrum not used for data. Alternatively, sync can be maintained by extracting the necessary clock frequency from the data signal itself.

Asynchronous systems are synchronized by a start signal, then a clock in the receiver continues the sampling rate for the forthcoming character bits. Noise can easily upset an asynchronous system if the start bit is dropped or a false bit is added.[22]

Channel direction. As well as being synchronous or asynchronous, channels are also characterized by their directionality. A channel which can only send or only receive is said to be simplex. Channels which can send and receive are said to be duplex. Of the class of duplex channels some can send or receive but cannot do both at once, these are called half duplex channels. Other channels can send and receive at the same time. These are called full duplex.

Two-wire versus four-wire. A slightly confusing nomenclature also exists in that the telephone companies use the terminology, two-wire and four-wire systems. The connection between the subscriber and the local central office need be only two wire, since no amplifiers or repeaters are required. These are unidirectional devices which make a two wire system simplex because of their unidirectionality. Thus lines requiring the use of repeaters must be four-wire so that communications can be full duplex.[23]

Transmission media. The transmission media for the signals may take the form of land-line, microwave or satellite.

Landline. Landline media includes twister pair and coaxial cable. All systems have some landline link to connect the subscriber to the system. Twisted pair has sufficient bandwidth to carry 24 voice channels. Coaxial cable is especially designed to control capitance and inductive effects and to shield the cable from extraneous signals; this as mentioned in the section on bandwidth provides a better frequency response, i.e. a higher capacity, higher bandwidth channel capable of a higher data rate. The bandwidth of the coaxial cable depends upon its design parameters.

Landline systems do not require carrier frequencies, in fact they would be dysfunctional. The only reason for a carrier frequency is to transmit data at a frequency so as to isolate the transmission from aberrant communications being transmitted through the same media, the atmosphere. Landline media is the wire which serves as the trough the carrier frequency serves in radio communications.

Optical fiber, laser and traveling-wave-tube systems are being proposed. But Punchard thinks that multiple coaxial cable and other conventional technology are a long, long way from being exhausted.[24] Traveling wave transmission is being used to interface the antannae with the amplifier-modulator/demodulator in microwave systems.

Microwave--line-of-sight. Microwave systems produce troughs within their line of sight. As frequency is increased the radio waves more and more resemble light waves in their ability to propagate in a straight line from their source. However, unlike light, the electromagnetic radiation is unaffected by clouds, rain or smog. Also, microwave antannae are designed to be highly directional to place maximum transmitting power in one direction. All microwave systems use frequency modulation to relieve the system of errors produced by multiple repeaters.

Tropospheric scatter. Tropospheric scatter systems are microwave systems in which the signal is beamed at the troposphere and scattered earthward over the horizon. A difficulty with the system is that the position of the reflecting troposphere varies with the hour of day and the level of sunspot activity. Usually this technique is used only to communicate with remote areas not reached by satellite communications. Though parties can communicate only a few hours a day, the cost of repeater station construction and manning in remote geographical areas is reduced considerably.

Satellite. Tropospheric scatter systems and satellite systems are made possible only by a technological breakthrough in high frequency amplifier technology. In both cases the received signal is relatively weak. Satellites cannot cost effectively house huge transmitters and the troposphere scatters only a small fraction of the transmitted signal earthward.

Landline and microwave are then the basic building blocks of data transmission systems.

Data Carrier Systems

Introduction. Three carrier systems which use the above transmission media and techniques will now be discussed: the Bell System, Western Union and Datran. The Bell System according to LaBlanc and Himsworth will continue to dominate the communications industry.[25] Western Union is building on its message capability and is currently overbuilding its microwave system. Datran (Data Transmission Company) filed with the FCC in 1969 to install a digital switched network to serve 35 cities from coast to coast. Canning concludes that the Bell System, Western Union and Datran will represent the main sources of data communications, but with the Bell System holding the lion's share.[26]

Western Union. Western Union has introduced a switched wideband service which allows the subscriber to select destination, bandwidth and mode of operation--data, voice or facsimile. A computer-switch data network based on the wideband service and the Western Union TWX-Telex system provide automatic switching with store and forward capability. Western Union has been the prime contractor in Air Force and combined services AUTODIN (Automatic Digital Network) and has gained expertise in proven message switching technology.[27]

Datran. Datran services rest on a network of 217 microwave repeater stations. Datran eventually will provide 9.6 or 19.2 kilobit per second channels capable of being subdivided to obtain 150 bit per second subchannels on a switched basis within the next two years. In addition 48 Kbps channels will be provided on a lease basis.

Other design goals for the Datran Network are an error rate better than 1 bit in 10^7 bits transmitted, connect time less than 3 seconds 99% of the time. A six second minimum charging time will prevail in six second increments.

Datran also expects to provide its own full duplex local distribution system from local district offices, via microwave links to the central city and cable from there to subscriber premises. The total system will provide end-to-end digital service with no conversion required.[28]

The Bell System. The future system. The American Telephone and Telegraph Company has recently set up a new Computer Communications and Data Services Division to meet the competition head-on. Bell has just initiated a new service called Digital Data Service (DDS) which it hopes to provide to more than 100 cities by 1977. DDS is a leased line service offering data rates of 2.4, 4.8, 9.6 and 56 Kilobits per second in full duplex.

At the local central office the various rates from all subscribers are multiplexed to form a stream of 8000, 8 bit bytes per second. The 8 bit byte includes 7 customer bits plus a control bit for the 56 Kbps rate and 6 customer bits, an empty bit and a control bit for the lower rates. These 8000 bytes per second are then multiplexed again in groups of 24 to form a 1.544 megabyte per second rate which is the capacity of one T1 pulse code modulation channel.

An exciting possibility is the Data Under Voice (DUV) system which will convert the present 500KHz unused bandwidth in AT&T's voice microwave system into a 1.544 megabyte per second data channel. This will mean that a digital data network can be provided to serve much of the United States with minor modifications to the present network. But with PCM channels Bell can go even farther and multiplex voice and data. Error rate will also be automatically controlled to 8 bits in 10^8 bits transmitted, a rate comparable to Datran. When errors in transmission are detected, DDS will notify the subscribers' equipment that errors have occurred but decisions on retransmission are the subscriber's responsibility.[29]

The present system. The above plans when implemented will change the Bell common carrier system into a digital network. The basic voice telephone service will remain the backbone of the entire telecommunications system.[30] A survey of 500 Computer Decisions readers who in total average 250,000 data calls per day indicate that the common carrier system leaves much to be desired. More than half the users rated service fair or poor. Results are tabulated below:[31]

U.S. Region	Common Carrier Data Service Good to Excellent	Fair	Poor
South Central	48%	40%	12%
Southwest	30%	42%	28%
Northeast	35%	45%	20%
Northwest	45%	50%	5%
Southeast	45%	45%	10%
North Central	40%	45%	15%

Yet, R.G. Canning concludes after studying common carrier service, that it presents a stable and pervasive base on which to build a telecommunications network.[32]

The telephone network is a hierarchical switched arrangement. To connect every telephone subscriber to every other subscriber would take on the order of 10^{15} wires if a hierarchical switching arrangement is not used. Each subscriber is connected to a local switching office, the local offices in turn connected to regional switching centers, and so on up to five hierarchical levels.[33]

The present telephone plant uses frequency division multiplexing to combine the basic 4 kilohertz voice channel into 12 voice channel groups, five group supergroups and ten supergroup master groups. Transmission of the FDM group hierarchy is usually single-sideband suppressed carrier.

A change being implemented now is a move to pulse code modulated transmission, a digital technique. At present signals originating from the subscriber are transmitted to the local central office over a standard two wire system. At the local central office these signal amplitudes are sampled 8000 times a second to the nearest of 128 different levels. The amplitude is coded in 7 binary digits with an additional bit added for signaling purposes. Eight bit bytes are then time division multiplexed to transmit 24 signals simultaneously. At the distant end the signals are separated and converted to analog voltages for transmission to a subscriber's telephone.

What has been described is the Tl carrier system which has a practical limit of about 50 miles between central offices. Transmission media is standard 22 gauge twisted-wire pair which is sufficient to transmit the 24 voice channel group at 64 kilobits per second with regeneraters needed every 6000 feet.[34] Installation of the Tl PCM carrier system began in 1961, mainly to expand short haul trunk capacity. By 1969, 250,000 miles of facilities had been converted to Tl systems, mainly in the Northeast United States.[35] A major advantage of this system is that it makes the common carrier system more reliable by eliminating crosstalk between channels, a major cause of data errors. As Canning describes it, PCM changes the telephone channel from a "trough" to a "pipe." The trough view is that if the circuit power in a channel becomes great enough, it can slop over into adjoining channels. PCM channels are pipes because they are insensitive to noise or circuit related voltage variations, since information is an on-off code.[36]

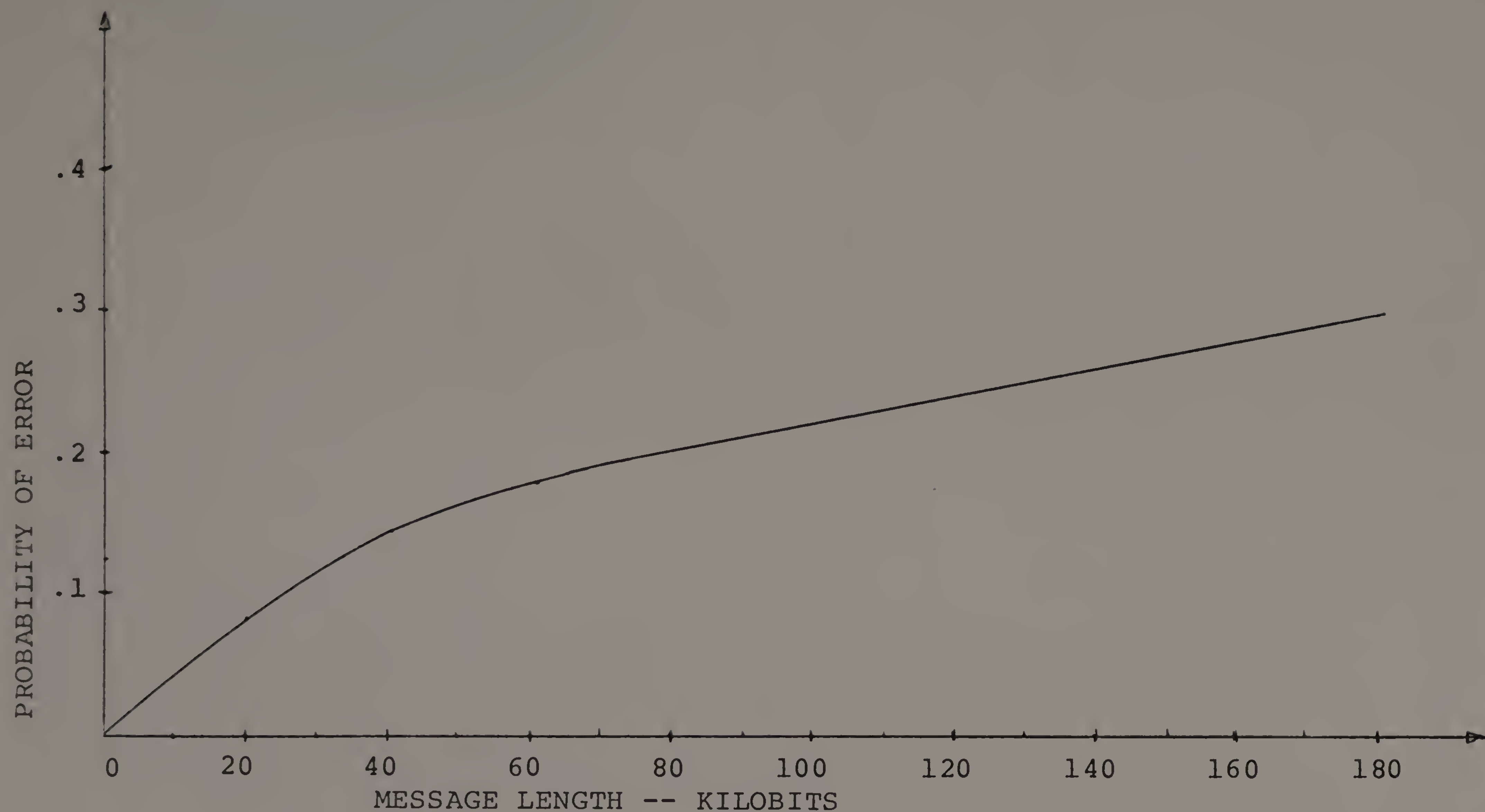
But T1 is not the end. The T2 medium haul system having a capacity of 6 megabits per second is now coming into service. One T2 circuit is equivalent to four T1 circuits. The T4 coaxial cable long haul system with 275 megabit rate is designed and a T5 system is under consideration. The fastest bit rate now being designed is 275 megabits per second. Time division multiplexers will interleave each bit stream into the next higher bit stream to provide a hierarchy of data rates to meet the needs of any user.[37]

Errors In Telecommunications Networks

Cause of errors. Errors in data communication systems are caused by impulses or voltage spikes, background noise or hum and the inherent nonlinearities in the system components and transmission lines. Impulse errors if not controlled become the major source of error in data communications. Pulse code modulation to some extent controls background noise and frequency compensation networks can be connected to lines to correct nonlinear frequency response.[38]

Impulse noise. In a system as diverse and complex as the telephone network there are many man-made and natural phenomena which introduce spikes or bursts of electrical noise onto communication circuits. Impulse noise tends to cluster. When things are bad, they are quite bad indeed. Two studies conducted by Bell Telephone Labs yield this conclusion--the Alexander, Gryb and Nast study in 1959 and the Townsend and Watts study in 1962.[39] This clustering tendency makes error detection more difficult because conventional parity codes may not recognize multiple errors in the same character or block. Approximately one-third of the time a character error occurs it will go undetected because of other errors in the same character.[40] A mitigating effect, however, is that clustering of errors often causes other surrounding characters to contain errors also and the faulty message or block will be rejected anyway. For this reason data in a long message accepted without parity error is less likely to contain hidden errors than data obtained by transmitting short blocks until they are free of error.[41] Figure 2 shows how message length relates to the probability of message error.[42]

Switching equipment a major cause of impulse noise. The switching equipment being used by common carriers is the major cause of impulse noise in data communication systems. Four types of switches are currently used: rotary, panel, crossbar and electronic. Rotary, panel and crossbar switches are electromechanical, their mechanical parts bounce and spark creating impulse noise upon contact.



PROBABILITY OF A MESSAGE BEING RECEIVED WITH ONE OR MORE ERRORS

Figure 2[43]

Rotary switches are the oldest and most numerous, especially in outlying, remote areas. Panel switches are still common in metropolitan areas and produce less impulse noise than rotary switches but are sufficiently error producing that they are avoided in conditioned data circuits. Crossbar switches may be used on data channels provided error detection-correction codes are used. The best switch is the electronic switching system (ESS) now being implemented in large offices. No impulse noise is generated, but the Bell System is only cutting over one office per day to this new system. At that rate it will be the year 2000 before all offices are converted.[44]

Given the above state of affairs the probability of getting a noisy circuit is high during peak periods when marginal equipment is brought on-line or in outlying areas where antiquated switching equipment is common. Another factor causing errors during peak periods is crosstalk. Janet L. Norman presents a study conducted by the Singer Company which found that the percent of error free calls at night was nearly twice that during daytime hours.[45]

An additional conclusion may be made, the greatest cause of data errors occurs within 20 miles of the source and destination points, because this is where antiquated equipment may be used.[46] The Singer study adds weight to this conclusion. The poorest performance was obtained in calls from New York City, only 30 miles from the New

Jersey receiver, while Denver, Colorado, achieved the best results.

<u>Background noise</u>. Background noise sometimes called white noise or Gaussian noise is almost a constant level. It is caused by the thermally induced random flow of electrons in circuits and is the common hiss sometimes heard during ordinary telephone conversation. Background noise tends to cause random errors involving individual bits. These individual bit errors are widely scattered in time and are hence easier to detect than impulse noise.

<u>Signal distortion</u>. Noise errors are heightened by signal distortions that occur because of the imperfect frequency response of the typical telephone channel. Distorted pulses are often indistinguishable as they become smeared together and stretched out in time. Nonlinearities in the line's distributed resistance, capacitance and inductance as well as nonlinearities in circuit elements are the cause of this imperfect frequency response. One way of reducing noise then is to use cable whose parameters are more closely controlled. Coaxial cable has a better frequency response than twisted pair because its parameters have been more closely controlled. Better frequency response lets it transmit the same data rate as twisted cable with more reliability, or a higher data rate (narrower pulses) with equivalent reliability.

To meet the problem of line amplitude and phase distortions automatic equilization techniques have been implemented. One of the more up-to-date devices is the transversal filter. It can be used as an adaptive equalizer when distortion is time dependent.[47]

The Arpa net error rate is one erroneous bit in 10^5 bits transmitted. Arpa uses error detection, retransmission and automatic alternate routing with the result that the equipment or any individual has never detected a single erroneous message that the data system has passed as correct. The Tymnet error rate is better than one erroneous bit in 4×10^{13} bits transmitted. Since these two systems use common carrier lines, we must conclude that the character of common carrier lines is not a problem for network growth.[48]

Error rates. Error rates are then determined by the character of the line and the switching mechanism used as well as the transmission rate. Noise at local central offices are critical because of the lower signal levels at that point. Canning says a telecommunications consultant he knows is of the opinion that no telephone circuit will have fewer than one error in 10^6 bits transmitted at today's nominal speed.[49]

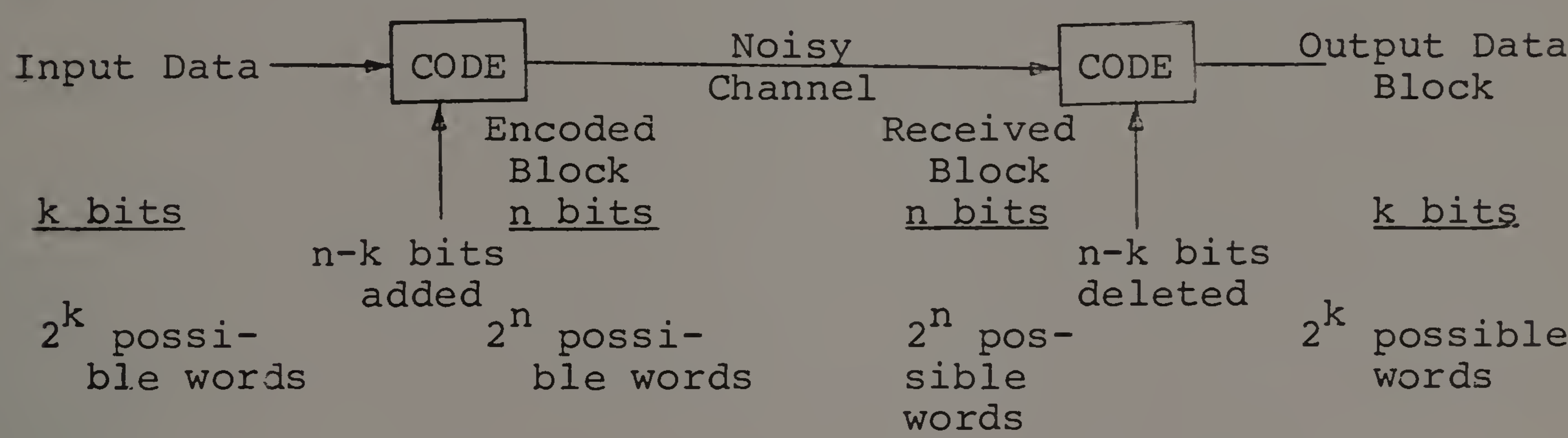
Error Detection And Correction

Introduction. The successful implementation of a data communication system is dependent upon its handling of errors. Errors must be detected as they occur and then corrected. We saw in a previous section that error rates are caused by burst noise or background noise. Shannon's 1948 "Mathematical Theory of Communication" sets an upper bound on information flow in any communications channel, provided that the proper encoding is found and used. One theorem says that with proper coding, any discrete channel can function up to its theoretical information flow limit with vanishingly small probability of error, in spite of the presence of noise! This finding was the impetus to search for efficient error-detecting, error-correcting codes.[50]

A variety of error detection and error correction codes are available. Some perform only error detection, requiring retransmission when errors are noted. These systems are sometimes called ARQ (automatic repeat request) systems. If the system is full duplex, the repeat request may be transmitted as data continues to flow. If the system is half duplex, line control discipline causes the direction of transmission to be reversed at the end of each block and the receipt of an error free block is either acknowledged (ACK) or not acknowledged (NAK). A good deal of emphasis is now being placed on forward error correction (FEC) which checks and corrects the message at the receiving station.

There is, however, no one best method of error control for all applications. A minimum-cost system satisfying a specific set of requirements can only be selected by considering the characteristics of the message and the error statistics of the transmission channel.[51]

<u>Theoretical basis</u>. The process of encoding binary data for error free transmission is a mapping of 2^k possible words onto 2^n possible sequences, where k is the number of bits in the input block and n is the number of bits in the encoded block (see Figure 3). The code requires n-k bits.

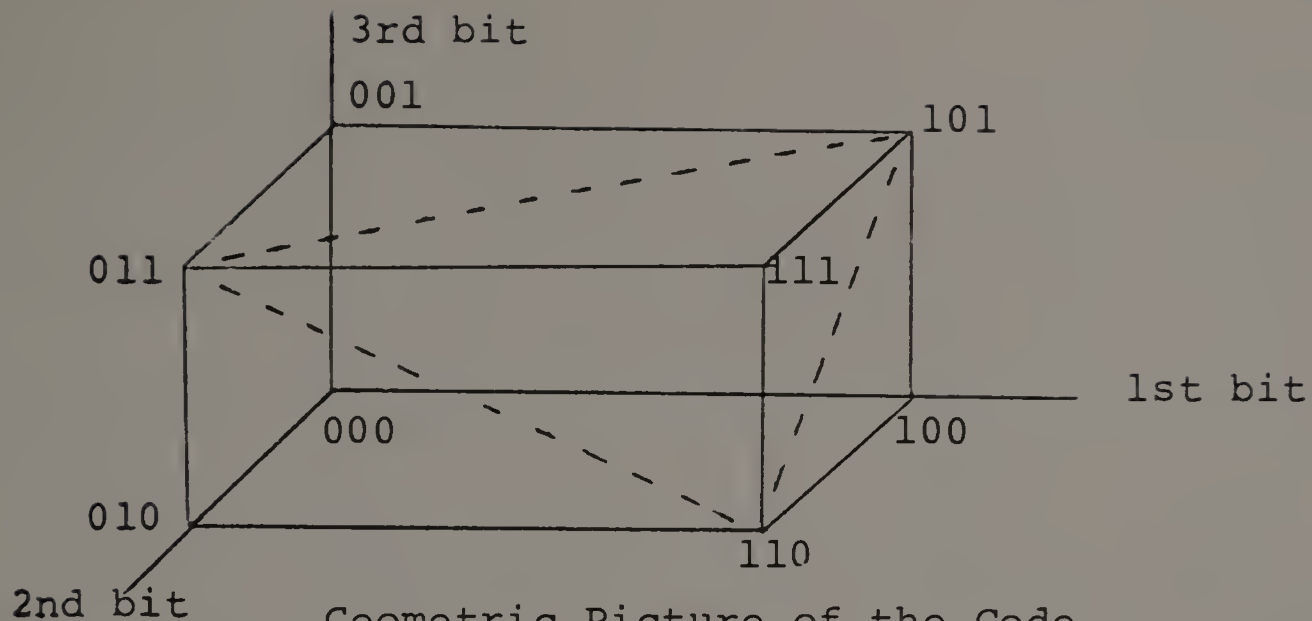


Error Control System

Figure 3[52]

If the input data block is 1 bit, then there are two possible information words. If two redundancy bits are added there are eight possible words of which only two are meaningful. The decoder then maps the received sequence into what it considers the most likely of the two information codes.

A geometric picture such as that depicted in Figure 4 is sometimes helpful.



Geometric Picture of the Code

Figure 4[53]

The three bits of the code are indicated along the three axes. The information bits are at 000 or 111. The plane connecting 011, 101 and 110 partitions space into two decision regions. Codes lying within either region will gravitate to the nearest extreme, 000 or 111; codes in the 011, 110, 101 plane will gravitate to 111. The probability of making a mistake is the probability that 000 is transmitted and the received sequence lies on or above the shaded plane, or that 111 is transmitted and the received sequence lies beneath the plane. Elementary probability theory can be used to evaluate this error.[54]

Convolution codes. Convolution codes provide a continuous bit stream of interleaved information and parity bits, encoded by shift registers and logic circuits according to a fixed set of rules. The parity bits are computed by a "sliding window" which uses a certain subset of the information bits within its aperture at a given moment.

The result of the encoding procedure is a code which can be described by a hierarchical tree structure. Each time a 1 or 0 is transmitted this gives a direction to move in the code tree.

Two basic methods are used for decoding convolution codes--threshold and sequential. The threshold method is easier to implement because it does not require the complex circuitry of sequential methods, however, sequential methodology is more powerful. It attempts to best fit the received data sequence with a tentative correct one postulated on the basis of past history of received data. When there is no immediate match, a search begins--backward and forward--along a prescribed coding tree. On the other hand threshold decoding manipulates the message and parity bits and computes an adjustment within a particular length of message.[55]

Convolution codes offer the best practical method for high volume (high data rate) forward error correction systems. This technique generates typical coding efficiency (ratio of information bits to total bits) ranges from 50 to 75 percent.[56]

Block codes. Constant-ratio codes. Block codes are of two main types, constant ratio codes and parity codes. Constant ratio codes use a block structure in which the ratio of 1's and 0's in each word is a constant. As long as a switch of a one to a zero and a zero to a one doesn't occur simultaneously, the code will detect the error.

Parity codes. A parity-check code relates a sequence of k information bits to a series of n-k parity bits, where n is the word length. Simple parity adds a single extra bit to the k-bit information block to make the number of ones either an even (even parity) or an odd (odd parity) number. This coding detects all single-bit errors, and all numbers of bit value errors.

Many schemes are available to derive parity bits from information bits, each with its own unique error protection properties. Two important types of parity-check codes are cyclic codes and geometric codes.

Cyclic parity codes. Cyclic codes derive their name from the property that any end-around shift of a word becomes another word. Each cyclic code is a group code, i.e. the code forms a mathematical group. Recall that a group is a collection of elements closed under an operation, that is, if two code words have any arithmetic operation performed on them another valid code word results. Thus, every polynomial in the code (for example 101101 is $x^5 + x^3 - x^2 + 1$) must be a multiple of some generator polynomial (for example, 100101 or $x^5 + x^2 + 1$). The binary message word is divided by the generator polynomial. The remainder, an encoding of the message word, is connected to the end of the message word and the concatenation transmitted. At the message destination the concatenation is separated into binary message word and remainder. The transmitted binary message word is divided by the generator polynomial and the remainder compared to the transmitted remainder, if a match exists, there was no error in transmission.

The code word is the encoding of the message word by the generator polynomial. It has been shown that the code property of minimum distance, i.e. the number of positions in which two code words differ, is simply related to the roots of the generator polynomial. Each error moves a word a distance of one. Decoding computational effort increases roughly as the square of the code length.[57]

Cyclic codes are made possible by the development of low cost shift registers and binary adders. The technique explained precisely is thus: (1) A number of zeros are added to k information bits. (2) The enhanced character, now n bits, is divided by a code generator polynomial of degree n-k by a n-k stage shift register. (3) The n bits and the remainder is transmitted. (4) At the distant end the enhanced character is again divided by a n-k stage shift register. (5) If the calculated and transmitted remainders are identical, no error has occurred in transmission.

The best and most useful class of cyclic codes is the Bose-Chaudhuri-Hocquenghem (BCH) code in which the roots of the generator polynomial are chosen in a certain manner to ensure a particular minimum distance between code words. BCH codes are capable of both error detection and correction. However, they are rarely used for correction, because of the extreme complexity of the decoder. Mainly, they are used in automatic repeat request systems, where these codes are capable of excellent error detection relative to coding efficiency for moderate code lengths. Code length is limited because their ability to correct errors does not increase in proportion to their length.

Gaushell generates curves of error performance of BCH codes for 31-bit and 63-bit codes for seven different combinations of n and k.[58] He concludes:

(1) The error-detecting power of a BCH code improves rapidly as the number of parity bits per word increases.

(2) The longer BCH codes provide better error detection for the same efficiency.

(3) Adding simple parity to a BCH-coded message further improves error detection. Simple parity detects all odd numbers of errors, which are not covered by the BCH code.

Geometric parity codes. Geometric codes are multidimensional blocked parity codes. They attack the deficiency of linear parity codes directly by extending the parity to more than one dimension. Sometimes called iterated codes, this technique has been known since the early days of coding and is used on most computer magnetic tape devices.

In addition to parity within a given word, geometric codes use parity between code words. For example, four words may be stored in succession at a transmitter. Then a fifth word is computed whose bits are the vertical parity checks on the columns of these words. Additionally, the diagonal may be parity checked. Figure 5 depicts such a scheme using odd parity.

0	1	0	1	0
1	0	0	1	0
0	1	0	0	1
0	0	1	1	0
1	0	1	1	1

Two Dimensional Geometric Odd Parity Coding

Figure 5

Note that this scheme allows complete specification of error location, and thus automatic correction. Decoding circuitry is also simple, requiring only a shift register and an exclusive OR.

Geometric code error performance is very good. Any odd numbered bit pattern is detected, because such a pattern always causes a vertical or a horizontal parity error. Also burst errors do not degrade the error-detecting capability of geometric codes significantly, since an undetected error would require identical bursts starting at the same bit position in an even number of data words.[59]

<u>Code concatenation</u>. A hierarchy of codes has been already hinted at with mention of the parity imbedded BCH

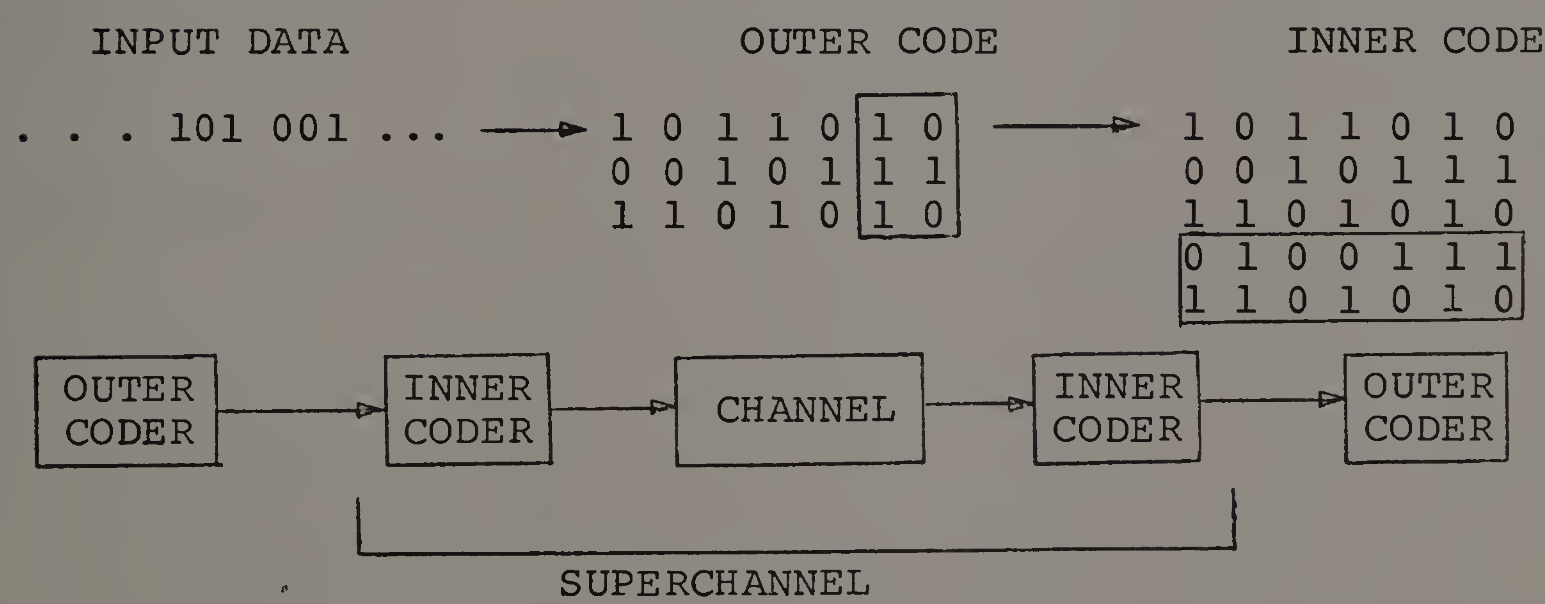
code and the previous geometric code. Elias in 1954 investigated a more general form of the geometric code. He found that as the number of hierarchical levels increased in number, i.e. the number of dimensions in a geometric code increased, the overall code rate (ratio of information bits to total bits) decreased, but by a decreasing rate. At infinity the code rate approaches some positive value and at the same time the probability of error approaches zero.

For ten years Elias' work lay dormant, then in 1965 Forney in the U.S.A., Ziv in Israel and Pinsker in the U.S.S.R. announced procedures capable of approaching channel capacity without exponentially growing complexity. Each procedure was based on a concatenation of known codes, the imbedding of codes within codes. Currently, all known schemes for realizing channel capacity without exponential growth of complexity involve concatenation.[60]

Concatenation uses an inner code to operate on digits to form blocks, an outer code uses these blocks as data, assembling and operating upon them with another code to form superblocks, etc. The concatenated coding system is analogous to the way redundancy is incorporated in language, letters forming words, words forming sentences.

The kind of code concatenation proposed by Forney is shown in Figure 6. Let's start on the inside of this system and work outwards with the description. The channel transmits digits one by one. We group these digits together to form words in which redundancy is incorporated. This is the inner code; in the example shown, the inner code consists of two redundant digits added to each 3-bit block. The input to such a coder is a three-bit block of information, 101, 111, etc. We can in turn consider these blocks as characters in another code, just as binary digits were characters in the inner code. This time the characters, instead of being binary numbers, are numbers chosen from a larger field. In this example the characters in the outer coder are chosen from a field of eight elements--the number of possible three-bit sequences.

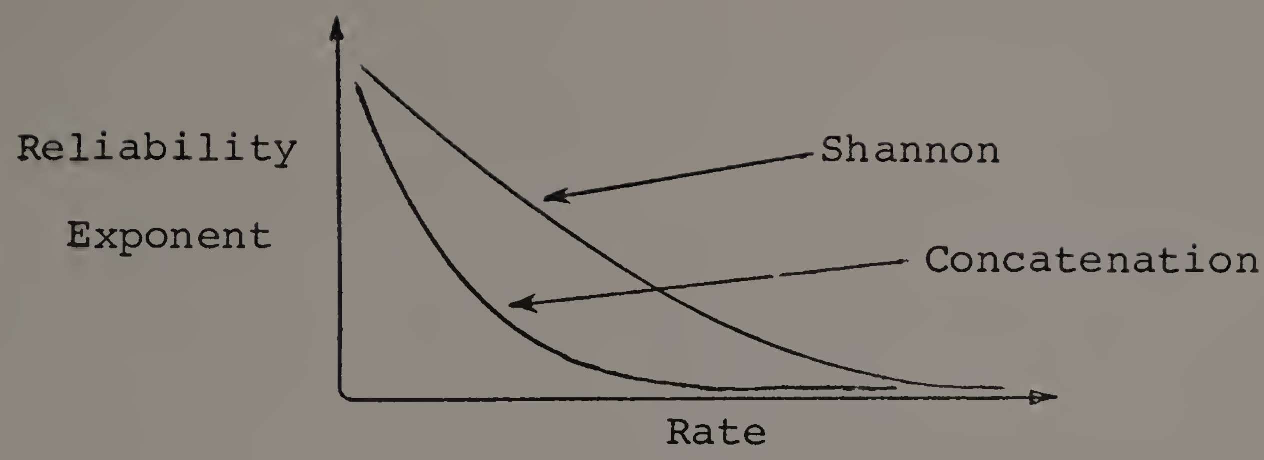
The outer coder groups these larger characters together and adds redundancy in the form of additional characters. In Figure 6 the outer coder takes 5 of the three-bit sequences, which act as its characters, and adds two additional three-bit sequences. The two additional three-bit sequences acts as checks; that is, they are computed in some fashion from the five initially specified characters.[61]



Concatenated Codes

Figure 6[62]

Concatenated codes have the advantage that their performance can be evaluated mathematically. Forney has proved a bound for concatenation similar to Shannon's bound. This boundary is depicted in Figure 7.



Reliability Exponent For Concatenation Coding Theorem

Figure 7[63]

The curve shows that by increasing the overall code length n the probability of error can be made increasingly small. However, the constant in the exponent of the concatenated coding bound is considerably smaller than in Shannon's bound. In effect a code must be much longer if it is formed by concatenating codes rather than constructing it in one fell swoop. But the exact penalty for hierarchical organization can be calculated.[64]

The reason concatenation loses efficiency is that the inner decoder destroys information in making its preliminary decision, thus decoding efficiency produces information loss.

Efficiency of concatenation relies mainly on the existence of a powerful algebraic code for the outer, superblock called the Reed-Solomon code. This code falls into the BCH code class, but unlike a BCH code, error correcting ability increases with length. This is the property which allows concatenation coding to reach the Shannon limit at large n.

Reed-Solomon codes are ineffective when used alone because the ratio of corrected bits to total bits is not constant with increasing n. Hence, asymtotic improvement cannot occur in channels where errors occur randomly. But if an inner code can be used to convert a channel from one in which errors occur randomly in bits to one in which errors occur randomly in characters the Reed-Solomon codes are then useful. The one difficulty is that the inner code must also be ideal in the Shannon sense, otherwise it will impute too many character errors to the outer decoder as its length increases.

Which code?. The conclusion as to which code to select can really only be determined by simulation using real channel data as Norman suggests.[65] It is well to realize, however, that Shannon's theoretical limit has not yet been reached and so errors will still occur and go undetected. Concatenation codes and convolution codes reach for the Shannon limit. However, in many applications the BCH-parity code combination is sufficient. Again, a proper match must be made between channel reliability, code reliability, data accuracy and integrity and the cost of the system versus the value of the data transmitted. The Data Administrator as the organization's data system engineer is responsible for ascertaining that the match is proper.

The next chapter will discuss the practical application of the foregoing theory to the data communication network.

FOOTNOTES

1. R.G. Canning, "Future Trends in Data Communications," <u>EDP Analyzer</u>, Vol. 7, No. 4 (April, 1969), p. 1.

2. R.G. Canning, "Current Status in Data Communications," <u>EDP Analyzer</u>, Vol. 7, No. 3 (March, 1969), p. 2.

3. H. Busignies, "Communication Channels," <u>Scientific American</u> (September, 1972), p. 99.

4. E.R. Kretzmer, "Communications Terminals," <u>Scientific</u> American (September, 1972), p. 135.

5. R.G. Canning, "Developments in Data Transmission," <u>EDP Analyzer</u>, Vol. 11, No. 3 (March, 1973), p. 4.

6. R.G. Canning, "Future Trends in Data Communications," <u>EDP Analyzer</u>, Vol. 7, No. 4 (April, 1969), p. 5.

7. R.E. LaBlanc and W.E. Himsworth, "Data Communications in 1980--A Capital Market View," <u>Proceedings of the Spring Joint Computer Conference</u>, 1972, p. 611.

8. <u>Ibid</u>., p. 615.

9. R.G. Canning, "Distributed Intelligence in Data Communications," <u>EDP Analyzer</u>, Vol. 11, No. 2 (February, 1973), p. 1.

10. E.B. Carne, "Telecommunications: Its Impact on Business," <u>Harvard Business Review</u> (July-August, 1972), p. 128.

11. R.M. Isavk, "Data Communications, State of the Market," <u>Honeywell Computer Journal</u> (Fall, 1970).

12. Adapted from M. Schwartz, "Information Transmission, Modulation and Noise," (New York: McGraw-Hill), 1959, p. 91 & 93.

13. Kretzmer, <u>Op</u>. <u>Cit</u>., p. 135.

14. In PM the phase of the carrier varies with the information signal and in FM the phase of the carrier varies with the integral of the information signal. See H.S. Black, <u>Modulation Theory</u>, (Princeton, N.J.: D. Van Nostrand Company, Inc.), 1953, Chapters 3 and 12.

15. R.G. Canning, "Future Trends in Data Communications," EDP Analyzer, Vol. 7, No. 4 (April, 1969), p. 5.

16. Schwartz, Op. Cit., p. 79.

17. Busignies, Op. Cit., p. 100.

18. H. Inose, "Communication Networks," Scientific American (September, 1972), p. 120.

19. R.G. Canning, "The Emerging Computer Networks," EDP Analyzer (January, 1973), Vol. 11, No. 3, p. 6.

20. Kretzmer, Op. Cit., p. 133.

21. H. Rudin, Jr., "Data Transmission: a direction for future development," IEEE Spectrum (February, 1970), p. 80.

22. P. Hersch, "Data Communications," IEEE Spectrum (February, 1971), p. 49-50.

23. H. Inose, Op. Cit., p. 120.

24. J.C.R. Punchard, "What's Ahead in Communications," IEEE Spectrum (January, 1970), p. 54.

25. LaBlanc and Himsworth, Op. Cit., p. 611.

26. R.G. Canning, "The Emerging Computer Networks," EDP Analyzer, Vol. 11, No. 1 (January, 1973), p. 8.

27. W.L. Harper, "The Remote World of Digital Switching," Datamation (March 15, 1971), p. 24.

28. R.G. Canning, "Developments in Data Transmission," EDP Analyzer, Vol. 11, No. 3 (March, 1973), p. 8.

29. Ibid., pp. 6-7.

30. Carne, Op. Cit., p. 127.

31. "Data Communications Survey," Computer Decisions (February, 1971), pp. 48-49.

32. R.G. Canning, "Developments in Data Transmission," EDP Analyzer, Vol. 11, No. 13, (March, 1973), p. 4.

33. R.W. Lucky, "Structure in Error-Correcting Codes," in Whyte, L.L., Wilson, A.G. and Wilson, D., eds., Hierarchical Structures (New York: American Elsevier Publishing Company, Inc.), 1969, p. 253.

34. Kretzmer, Op. Cit., p. 138.

35. R.G. Canning, "Future Trends in Data Communications," EDP Analyzer, Vol. 7, No. 4 (April, 1969), p. 4.

36. R.G. Canning, "Current Status in Data Communications," EDP Analyzer, Vol. 7, No. 3 (March, 1969), p. 7.

37. Kretzmer, Op. Cit., p. 138.

38. P. Hersch, "Data Communications," IEEE Spectrum (February, 1971), p. 51.

39. A.A. Alexander, R.M. Gryb and D.W. Nast, "Capabilities of the Telephone Network for Data Transmission," Bell System Technical Journal, Vol. 39, 1960, p. 431 and R.L. Townsend and R.N. Watts, "Effectiveness of Error Control in Data Communications Over the Switched Telephone Network," Bell System Technical Journal, Vol. 43, 1964, p. 2611.

40. J.L. Norman, "Reducing Telephone Network Errors," Datamation (October, 1971), p. 26.

41. Ibid., p. 27.

42. Ibid., p. 26.

43. J.L. Norman, Op. Cit., p. 26.

44. R.G. Canning, "Current Status in Data Communications," EDP Analyzer, Vol. 7, No. 3 (March, 1969), p. 5.

45. J.L. Norman, Op. Cit., p. 30.

46. R.G. Canning, "Developments in Data Transmission," EDP Analyzer, Vol. 11, No. 3 (March, 1973), p. 4.

47. H. Rudin, Jr., Op. Cit., p. 80.

48. R.G. Canning, "The Emerging Computer Networks," EDP Analyzer, Vol. 11, No. 1 (January, 1973), p. 11.

49. R.G. Canning, "Current Status in Data Communications," EDP Analyzer, Vol. 7, No. 3 (March, 1969), p. 8.

50. Kretzmer, *Op. Cit.*, p. 131.

51. D.J. Gaushell, "Error Control in Digital Transmission," *Control Engineering* (June, 1972), p. 59.

52. Adapted from R.W. Lucky, "Structure in Error-Correcting Codes," in Whyte, L.W., Wilson, A.G. and Wilson, D., eds., *Hierarchical Structures* (New York: American Elsevier Publishing Co., Inc.), 1969, p. 256.

53. Lucky, *Ibid.*, p. 258.

54. *Ibid.*, pp. 256-258.

55. Hersch, *Op. Cit.*, p. 52.

56. Gaushell, *Op. Cit.*, p. 59.

57. Lucky, *Op. Cit.*, p. 264.

58. D.J. Gaushell, *Op. Cit.*, p. 62.

59. Gaushell, *Op. Cit.*, p. 63.

60. Lucky, *Op. Cit.*, p. 269.

61. Lucky, *Op. Cit.*, pp. 269-270.

62. Lucky, *Op. Cit.*, p. 270.

63. Lucky, *Op. Cit.*, p. 272.

64. Lucky, *Op. Cit.*, p. 273.

65. J.L. Norman, *Op. Cit.*, p. 27.

CHAPTER V

DATA COMMUNICATIONS SYSTEMS DESIGN AND IMPLEMENTATION

Introduction

The preceding chapter examined the theoretical grounding the Data Administrator must have in data communications in order to assist and control the design of the data communication network as it pertains to data accuracy, integrity and security. This chapter applies the theory developed previously to data communications system design and implementation, and like the chapter on input systems develops a set of application environments which can be used by the Data Administrator.

The man-computer link. The important link between man and computer-based information system is the data communications network which is why it is such an important component of the MIS. Terminals connected to network endpoints bring the use of the computer to the place where the data originates, to people at remote locations. The information system can get fast reporting from sources never before possible to reach except by mail and then put summarized decision data at other remote points.

The Chicago-based Swift meatpacking company, for example, uses its grographically distributed information system for applications in accounting, financial information, sales and price and order data entry, etc. One application in particular takes information from each pork plant location on every hog killed. A ten-digit number encodes the hog's lot number, grade, and weight. All 75 teletype terminal locations in several plants are polled at least five times every hour by a central computer. One paper tape message is read every time a terminal is polled. During the evening hours the data is analyzed and decision information transmitted to the plants.

The plants are told which weights and pork grades currently represent the best buy based on market conditions. Additionally, plants get data on what type of yields they can expect from the hogs butchered the day before. The meatpacking business is a high volume, low margin business and the data tells the buyer where to look for the best buys and it tells the plant manager whether the buyers have done a good job grading live hogs. The computer system operates twelve hours per day transmitting from 2000 to 2500 messages from Chicago to outlying areas every day.[1]

So, even a not very sophisticated network can produce huge benefits. In another application Jim Brown of the First National Bank of Atlanta calculated the dollar cost of physically trucking documents to be greater than the cost of communication lines. The shorter data sensing time was only an additional plus of the system.[2]

Thus, we can see that as computer-based information systems evolve, the communications component will become more and more important. J.T. Hootman sees further, he sees the formation of computer networks and the interconnection of these networks, just as railroads were connected, power utilities, telephone networks, etc. Both Strefferud and Hootman see these networks becoming market places, where suppliers of propriety services will offer these services via the networks.[3]

A biological prespective. The proper communications system prespective can be gotten by examining a biological system. In higher level organisms each bundle of nerve fibers has its own characteristic conduction speed. For example, the signals conveying muscle position travel at the highest velocity, presumably because balance and quick movement are absolutely vital, whereas pain signals are among the slowest. One cannot doubt that the varying transmission speeds represent an optimization of the total biological communication system, achieved over millenniums by progressive evolution.[4]

Fail-soft operational level communication vital to organizational balance. The information system too, must be well balanced. Operational level data must be sufficiently responsive to allow the organization to maintain its balance and to survive. The tactical and strategic systems which will use more data generated outside the organization are more likely to use data and resources contained in commercial computer networks. Higher level systems must, however, be built upon a solid operational base. Organizations are now intent upon designing this operational base.

Most of the comments to follow then are especially applicable to implementing geographically distributed operational level systems. The data base may be also distributed at remote locations, so that operational level data is available to allow continued functioning when the communication link is out of order. In a like manner remote terminals will contain logic so that operational data can continue to be collected when the communication system fails. This fail-soft system flexibility is the theme of this chapter on data communication systems design and implementation and the basic theme running through Data Administration itself. The information system puts data at decision points within the time and accuracy constraints necessary to maintain system control. The data administration function ensures the data's accuracy, security and integrity and ascertains the data will be there within the constraints irrespective of system degradation.

Reason for communication protocol complexity. The biological analogy of the communications system works quite well, but for a complexity nonexistent in any other system. This complexity derives from the fact that control information is intermixed with data. This sharing of the same facility for data and control is very unnatural and leads to some quite elaborate protocols to separate the two. The unnaturalness is made poignant when one considers that computer logic is simplified and faster if data buses are distinct from control lines and software logic separates procedure from data structure (the data base concept). The reason for control and data channel sharing in telecommunications is evident, the cost of an extra channel would be immense compared to the additional complexity of deciphering control from data.[5]

Other telecommunications system design interference is due to the fact that computer and telecommunications technology is rapidly changing, and when changes in one area occur it is often necessary to rethink the technology used in other parts of the total system. Hence, the design emphasis is on flexibility to allow system segments to be upgraded or completely changed.

A babel of communication codes. Another confounding event is the Babel of Communication codes in use today. However, two codes are coming to the fore, ASC II (the United States of America National Standards Code for Information Exchange) and IBM's EBCDIC (extended binary-coded decimal interchange code).

ASC II was developed in the early 1960's to be a single comprehensive code that was intended to become the standard. Today, it is the standard of the data communications industry, but not yet the computer industry standard. IBM did not like the 128 character limitation imposed by a 7-bit code, so it used the 256 possible character 8-bit EBCDIC in its System/360 series. Other computer manufacturers followed, with the notable exception of the NCR (National Cash Register) Century series constructed around an internal ASC II code. The result is that computer overhead must be wasted making unnecessary code conversions between internal computer data representation and external communication system data representation. Canning sees it as a case of the irrestible force (government) meeting the immoveable object (IBM).[6]

Distributed Intelligence in Data Communications

Introduction. First generation communications controllers used a portion of the main CPU to control communications. As teleprocessing grew, a CPU upgrade was necessary to meet the needs of the teleprocessing function. Some CPU's spend from 40 to 60 percent of their capacity on the teleprocessing function.[7] However, the type of processing required of the CPU for this function is wasteful of valuable resources. Users are discovering that relegating this function to a minicomputer front-end is cheaper, more reliable and adds a degree of flexibility to the system.

Minicomputers do not have the complexity of software or hardware in the mainframe and therefore are more reliable.[8] Minicomputer core generally is one-half the price of mainframe core and "failsoft" flexibility is enhanced because the dispersal of front-end responsibilities make it more unlikely that the user will lose his entire communications functions. Dual telecommunications minicomputers may be cost effective, or as Dan Zatyko of Varian Data Systems sees it, the front-end responsibilities may even be dispersed among a multitude of specialized minicomputers.

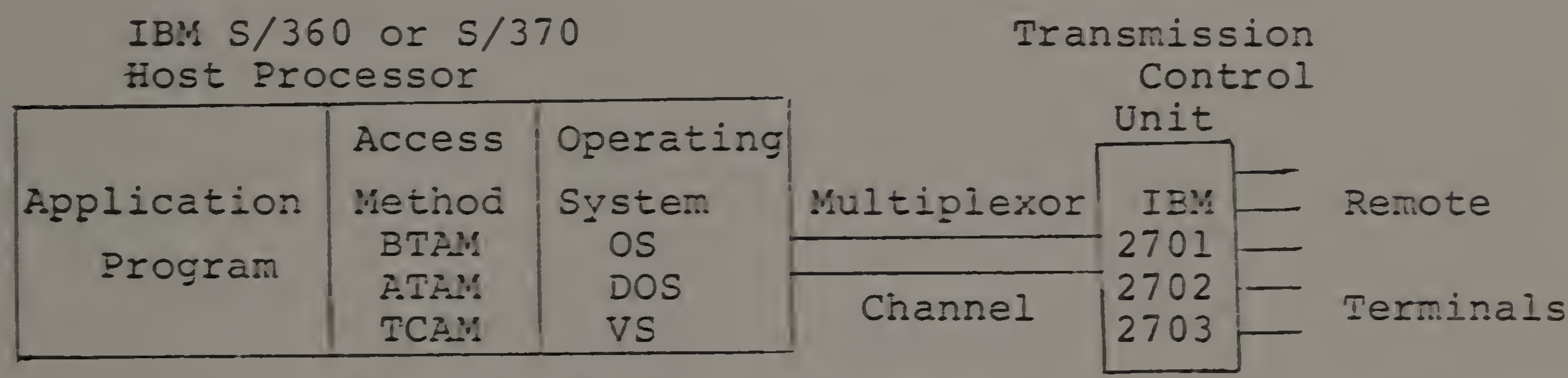
Phil Cleveland of Tempo Computers, Inc. goes one step further and suggests that the front-end take over many repetitive I/O bound DP tasks, such as printing and the sort/merge operation. Complete software assemblies may even be moved to a satellite processor where specialized DP operations can be run concurrently with teleprocessing programs. Daniel Sinnot, president of Interdata, Inc. sees a marriage of industrial control with data communications as remote industrial monitors transmit their data directly to the communications front-end.[9] Such high flying though reasonable projections set the scene for the more mundane topic of analyzing the costs and benefits of the centralized versus the decentralized approach in telecommunications processing.

Centralized telecommunications intelligence. In the centralized telecommunications approach the communications control program (CCP) is resident in the mainframe CPU. The communications control program is designed for specific terminals, line speeds, communication protocol, etc. In addition the CCP is intimately tied to the mainframe operating system. Also, all remote terminals must be serviced through this one program leading to the probability that a faster CPU may be required.

The types of data communication procedures handled by the CPU resident control program include: (1) maintaining a list of all terminals serviced (security measures may be included here), (2) identifying the source of each message (terminal location possibly recorded for security purposes), (3) message construction from characters and bits, (4) code conversions from terminal code to internal machine code, (5) error detection, correction, acknowledgement and retransmission techniques.[10]

In addition the centralized approach requires a hard-wired, specialized line controller called a communications control unit (CCU). The CCU is intimately tied to the CPU-based communications control program, the transmission line characteristics and the type of terminals serviced. The CCU buffers incoming data temporarily but has storage for only a few characters. The CPU must then be frequently interrupted.

The result of the centralized intelligence approach is a rigid system. Line speeds are difficult to change, line protocol is difficult to change and the user is limited to the types and mixtures of remote terminals he can use. Examples of this type of configuration is the IBM System/360 and System/370 2700 series communication control unit in conjunction with the Btam (basic telecommunications access method), Qtam (queued telecommunications access method), Tcam (telecommunications access method) communication control programs.[11] (see Figure 1) These access programs consume from 75 to 200 kilobytes of main core depending upon the number of communication lines used.[12]



IBM TELECOMMUNICATIONS SYSTEMS

Figure 1[13]

There has been a move to replace conventional IBM front-ends with emulators, true communications processors or intelligent emulators. The approach has been to simulate an IBM telecommunications processor with a minicomputer. The emulator gives the user a "security blanket." The system is unchanged, only hardware costs are saved. The front-end often can "fool" the IBM software by making an unsupported terminal appear to be a supported terminal. The real savings occur, when the user moves the entire communications function out of the mainframe![14]

One unusual emulator is the Digital Equipment PDP-11 based DX-11b. The system is based on the IBM Graphics Access Method (GAM) which requires only 5 Kbytes of mainframe core. This makes the user less vulnerable to software changes since the front-end acts like a hardware controller. The system can emulate all IBM front-ends and access methods with special enhancements.[15]

Distributed telecommunications intelligence. Recently, with the 3705 IBM has begun to move some intelligence out of the CPU. But the 3705 also has limits, it cannot support 300 bps terminals (no IBM terminal operates at that speed), it cannot support the varied codes and communication procedures used by independent terminal manufacturers and it cannot support CPU-peripheral interfacing with tape, disk, drum, card reader or line printer.[16]

Functions and configurations of a Front-end processor. The functions of a front-end telecommunications processor have been enumerated in a recent Datapro Research Report. The research report says that some or all of the following functions could be included depending upon the application environment:[17]

(1) Line Control, terminal polling and automatic call answering.

(2) Character and Message Assembly, message formatting to be compatible with the mainframe.

(3) Data Conversion into the machine code of the host CPU (ASC II or EBCDIC).

(4) Data and Message Editing, data compression and other techniques for faster transmission.

(5) Error Control to keep incorrect messages from entering the main processor.

(6) Message Buffering and Queuing to allow data to be fed into the host CPU at a compatible rate.

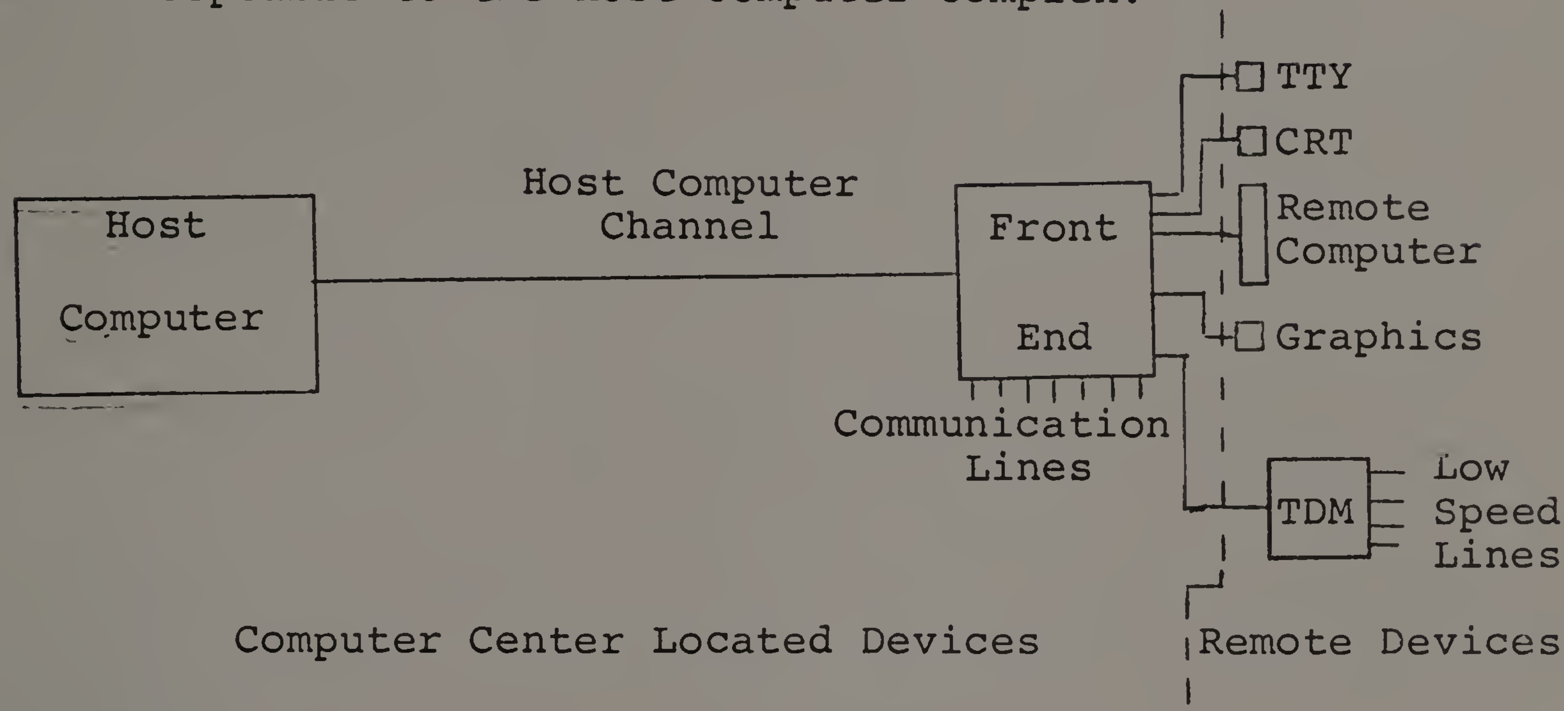
(7) Message Switching, an important function when the front end is connected to several mainframes.

(8) Message Answering, including replies which do not have to tie up the host system.

(9) Message Recording, an important function when a network failure occurs.

(10) Statistics Recording to keep a record of traffic, errors and other housekeeping functions.

These functions are executed within a minicomputer in one of five basic configurations which are based on software and electromechanical considerations of the user's mainframe applications.[18] The first type of front-end system is plug-compatible, connecting physically and electrically to the host computer channel as though it were a standard peripheral (see Figure 2). A characteristic of this front-end is that it emulates a well-defined subset of the devices it replaces, for example, the 270X and Btam, and performs additional functions such as code conversion and character to message assembly, etc., unavailable to users of Btam. The biggest advantage though is dollar savings. A 128 terminal application replacement can save $40,000 to $100,000 annually. In addition response time is often improved because less data manipulation and drum accesses are required. Another flexibility plus is that savings can be realized by letting the front-end perform time division multiplexing, thus reducing the number of TDM adapters required and halving TDM costs. Another feature is that computer-based message concentrators, non-compatible host computers, non-supported terminals, and TDM equipment can be made acceptable to the host computer complex.

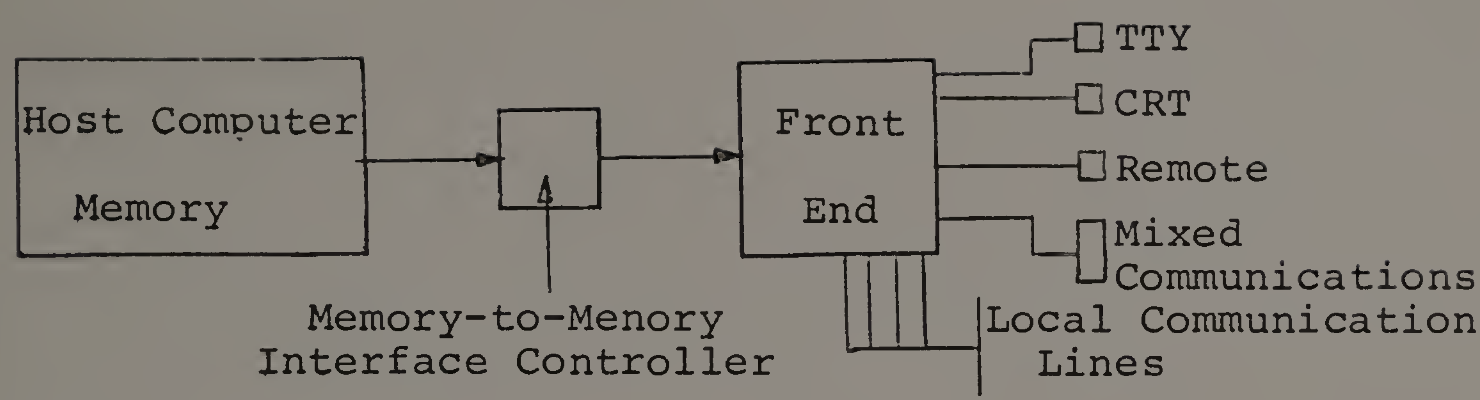


PLUG COMPATIBLE FRONT END (Emulation Plus)

Figure 2[19]

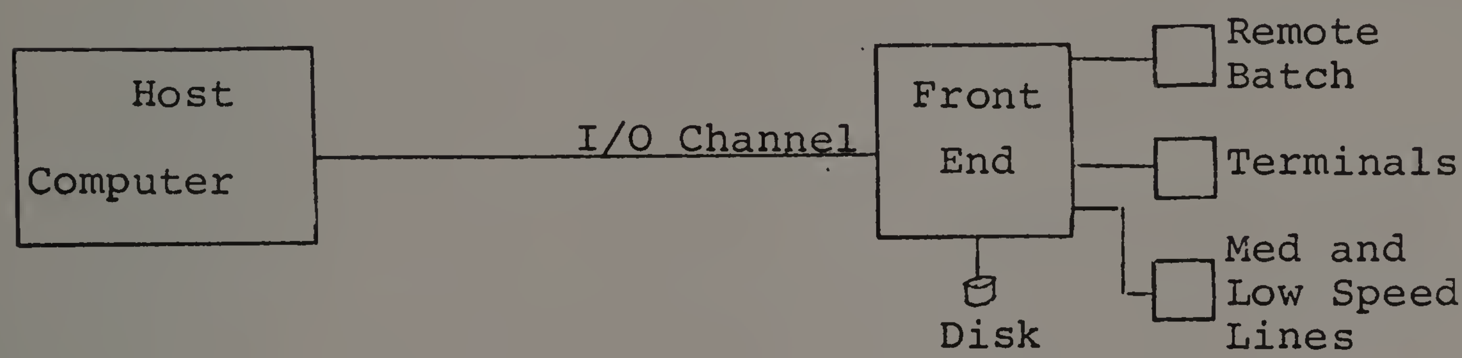
The second configuration is the core-to-core front-end used in large systems where a fast core transfer system is required (see Figure 3). This configuration has limitations based on the interconnection. Since computers are interfaced via common core, both computers must use the same character codes and memory control signals.

The third configuration makes the communication front-end act like a standard magnetic tape or drum peripheral (see Figure 4). This configuration requires a sophisticated mainframe operating system which will allow the application program to communicate with a wide variety of peripheral devices in a standard way. The disadvantage of this approach is that the front-end and mainframe almost always have to be



CORE-TO-CORE FRONT END (High Speed)

Figure 3[20]

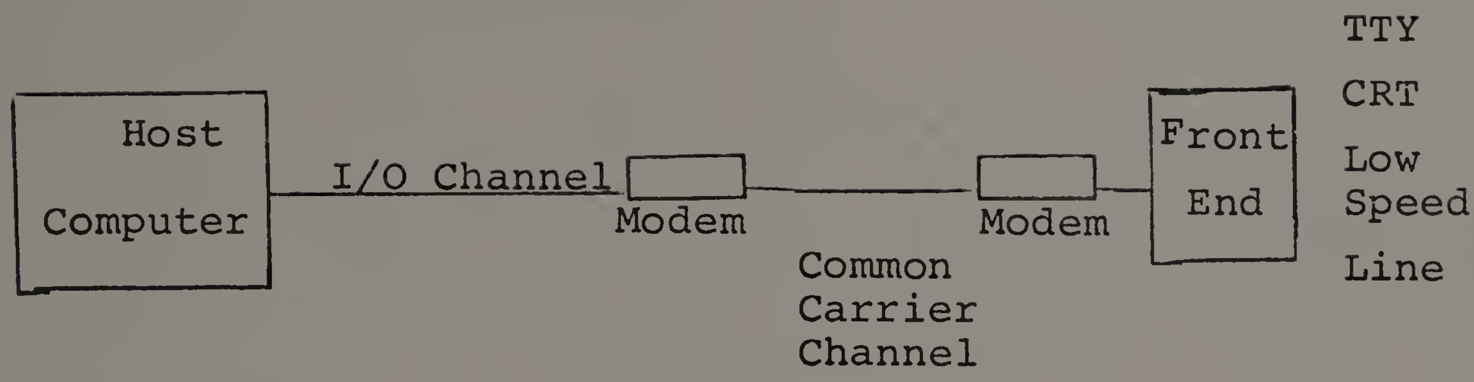


A VIRTUAL I/O PERIPHERAL FRONT END

Figure 4[21]

be of the same manufacture. However, the advantage is software compatibility between mainframe and telecommunication applications.

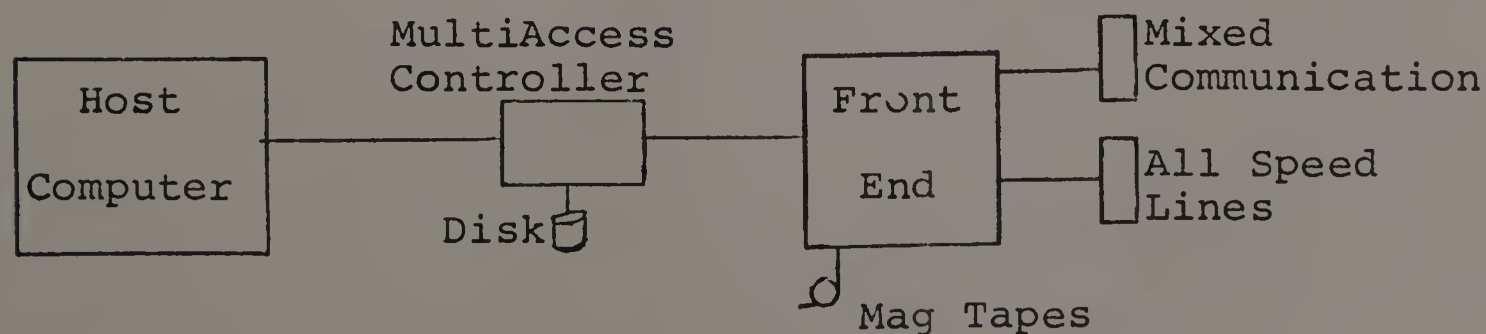
The fourth configuration uses a data link between the front-end and the mainframe (see Figure 5). It is the best approach for multicomputer systems. Disadvantages obtain because additional communications equipment must be purchased from each computer vendor and each port requires this equipment.



FRONT-END WITH DATA LINK PROCESSOR CONNECTION

Figure 5[22]

The fifth configuration has an intermediate direct access storage device to transfer data from the front end to the host computer (see Figure 6). In this configuration the mainframe and front end interface by interrogating fixed control areas of the DASD for information transfer instructions. An alternative is to add a high speed processor-to-processor interrupt line to provide an instruction and command path. The chief advantage of this approach is that each system can recover from a common file and high data transfer rates can be used when the data is available and standard operating system software can be used in the mainframe.



FRONT-END WITH INTERMEDIATE STORAGE LINK

Figure 6[23]

The above configurations are but a sample of the multitude of possibilities. Jon Gould, director of data communications at Interdata, however, considers these five configurations to be basic to all the possibilities.[24]

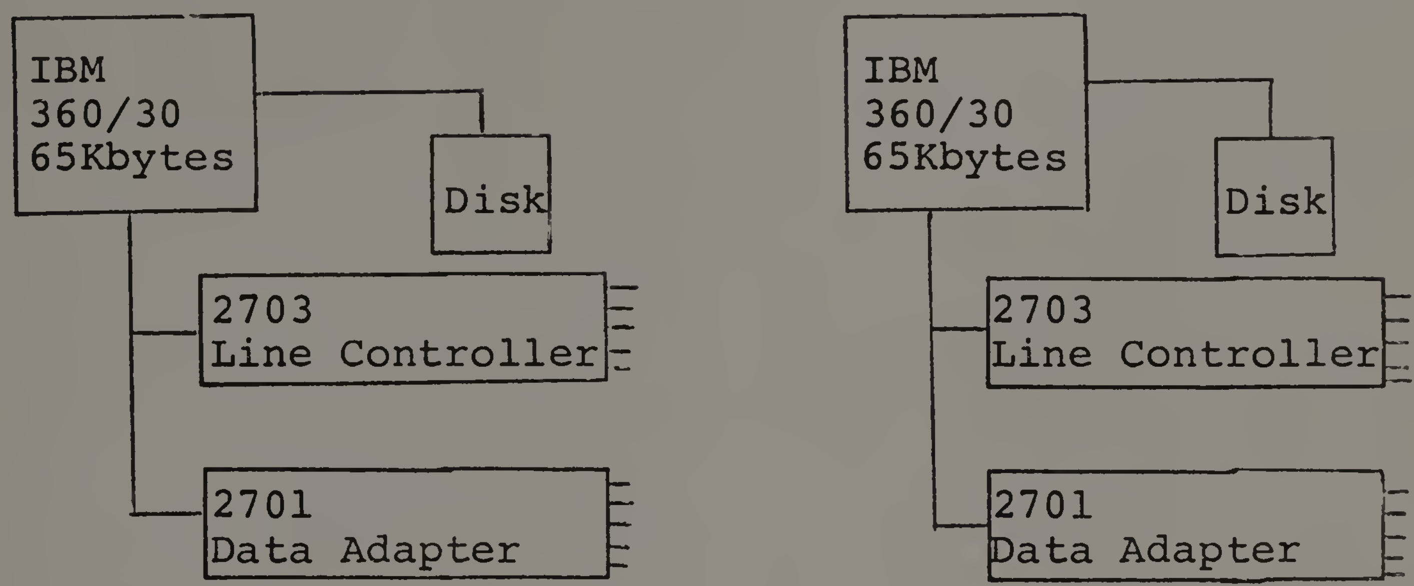
Decreased complexity and savings of the distributed approach. The increasing complexity of the computer communications system is the reason communications system intelligence is being moved out of the mainframe. When multiple communication lines come into a DP site, the user must service these lines. Often the system has hundreds of lines of differing speeds, differing bit formats (codes), differing parity checking schemes and differing error detection/correction schemes. The functions having to deal strictly with communications have to be handled before the data can be processed by the mainframe. It is logical and cost effective to move the telecommunications processing out of the mainframe. Often sufficient mainframe code is released that an upgrade to a larger system may be postponed. Phil Cleveland, manager of systems marketing for Tempo Computers, Inc., says that replacing an IBM 2703 100 terminal system with a 270T system from Tempo could save from $2000 to $5000 per month, and if the user had a 360/50, the core savings could forstall mainframe upgrade to a 360/65 or 370/155.[25]

Champion Service Company recently moved its communications logic out of its central computer and saved more than

$250,000 per year. Champion is a service company providing accounting services to more than 35 Savings and Loan companies having over 300 remote terminals ranging in speed from 134.5 to 1200 bps. The original system depicted in Figure 7, had dual 360/30's, twin 2701 data adapters renting for $50,000 each per year and twin 2703 line controllers renting for $125,000 each per year. The system was replaced with a PDP-11/20 renting for $100,000 per year, a savings of $250,000 per year, excluding rental on the 360/30.[26]

The new system is depicted in Figure 8. The old system processed 60 to 150 transactions per minute from 700 accounts served by 300 terminals. The new system can process 60 to 200 transactions per minute from 1 million accounts served by 400 terminals. This system upgrade resulted even though one CPU was dropped and one 24K word minicomputer with disk was added!

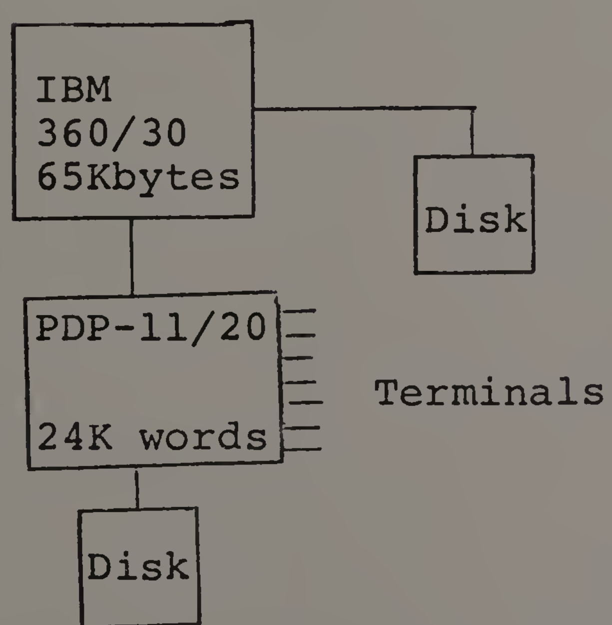
The system constructs a work-in-process file in the PDP-11 when the teller inserts the passbook and keys in transaction data. The data is later read into the 360/30 where the transaction is totaled and information is used to update the master file. Additionally, any teller cash drawer can be balanced at any time. The system has experienced an up-time of from 98 to 99 percent. It was deemed appropriate that a full time person interface with the common carrier on line problems. Another 360/30 is planned should the number of terminals exceed 400 or should system response time exceed 12 seconds. System response is presently 6 seconds.[27]



30-75 transactions/min	30-75 transactions/min
350 accounts	350 accounts
150 terminals	150 terminals

CHAMPION SERVICE COMPANY--OLD SYSTEM

Figure 7[28]



60-220 transactions/min
1,000,000 accounts
400 terminals

CHAMPION SERVICE COMPANY--NEW SYSTEM

Figure 8[29]

So we see a concrete example of how moving intelligence out of the mainframe can add extra system capacity and decrease response time while saving money.

Intelligence in switches and node switches. The front end processor is only one segment of the distributed intelligence system. Other minicomputers may be used throughout the system as a switch in a centralized star structure of remote computers or as a node switch in a computer network. The Arpa net is an example. Intelligence here is used in store-and-forward message applications and message blocking. These units may also assume network control, handling noise and line outage problems by automatically switching out and reporting bad lines. Messages may also be automatically switched to the proper destination as in AUTODIN, the armed services Western Union designed automatic switching network. The minicomputer may also act like a multiplexer or a concentrater combining lines for high speed transmission. An important aspect is to monitor network errors; block error counts, and counts by line or by terminal can isolate faults and be used to determine alternate routing.[30]

Intelligence in terminals and terminal controllers. Intelligence may also be integrated into the terminal or the terminal controller for a group of terminals. The data input chapter discussed minicomputer-based intelligent terminals and their system input editing and stand alone capabilities. This "fail-soft" capability can also be found in minicomputer-based terminal controllers. Data may be accepted for a time while communication line difficulty or system failure precludes full operation. Intelligent terminal controllers may also act as multiplexers and concentrators combining many low speed lines for transmission over a high speed line. Unattended terminal operation and data error and validation checks can be performed by the controller to catch errors before they enter the network.[31] The intelligent terminal controller can act as a sink for incoming messages, holding them for distribution and as a source of error-free, concentrated network data.

Data Communications Systems Planning

Network processing versus information processing. The design and planning of a data communications complex must be integrated with the design of the information system as discussed in Appendix A. Design is further simplified if network processing is separated from data processing. Network processing refers to data flow between computers or between sensor and computer. Data processing concerns itself with data manipulation to produce information as well as data base structuring. Together, network processing and data processing form the information network.

General Foods Corporation recently studies their information network and concluded that data processing planning should be separate from network processing planning, because data processing applications change more rapidly than data communication services. By viewing their network processing applications General Foods saw a need to integrate network processing planning. Heretofore, they were using a number of different lines, services and terminals. Savings and efficiency increases could be obtained by integrating network activities.[34]

Five planning parameters. As stated previously the network processing and information processing functions must be examined jointly on a macro scale in the initial system conception. Once the system sensors and decision points are located and analyzed along five dimensions, the information processing requirements may be detailed and the connective network processing requirements may be detailed separately. The Data Administrator has operational data responsibility in both information processing, as it pertains to data base manipulation and security, and network processing, as it pertains to data integrity, accuracy and security.

The five parameters upon which to examine the information network are identified by Becker to be: (1) geographic dispersion, (2) volume of information transmitted, (3) the amount of data manipulation required, (4) the decision point response requirement and (5) the network availability.[35] Geographic dispersion may require cross country, worldwide or interplant communication links. Data volume transmitted along these links and the time pattern of the communication will specify line capacity and quality. We saw in the previous chapter that data transmission errors were dependent on the time of day the transmission occurred. The amount of data manipulation necessary at remote points and data importance to continued organization operation will determine whether data processing should be performed at a central site or remotely as well as where the data base should be centrally located or distributed throughout the network. If critical operational data bases are located at points of operational level control, then the organization can continue to function even though a network link is inoperative. Additionally, system response is enhanced because of shorter lines of communication. System response at the tactical and strategic levels is not as critical, response time being measured in days or weeks rather than minutes or hours.

Network availability refers to a balanced design so there are no ungovernable overloads. There must be sufficient alternate data links and data "sinks" throughout the network to buffer the flow of data. A data "sink" might be an intelligent terminal controller or a data switch whose function is to maintain a constant maximum data flow through the scarce resource communication network. The communication links are to network processing what the CPU is to information processing, it is a scarce and expensive resource which must be maximally utilized. All network processing techniques exist to maximize data link use. Data link optimization techniques will be discussed in the next sections.

<u>Network optimization</u>. Data link structure may be optimized by using one of several optimization packages available. For example, one of the earliest packages developed was IBM's Communications Network Design Program (CNDP). Its successor is called TP net. These particular programs and their details of operation are proprietary and only for the use of IBM salespersons. However, a recently published article demonstrates the complexity of CDNP:

> The package accepts input data in the form of tariff information, terminal and computer locations, terminal traffic volume, and line loading and response needs. The IBM technique assumes that a star network of point-to-point links from every remote terminal to one of the central (CPU) complexes is the most costly (worst case) layout.
>
> In successive iterations, it removes a link that attaches to one of the centers, and replaces it with some other shorter link to reduce line costs by the formation of multipoint lines. At each iteration the method selects a particular pair of links (one to be removed and the other added) to produce the greatest net reduction in link cost without violating line traffic constraints.
>
> Although this method takes care of traffic overloads, it assumes the placement of concentrators and disregards the use of mixed grades of service in the same network.[34]

This and other packages are based on a coordinate system developed by Bell which establishes V and H coordinates through all rate centers within the U.S. phone system. Network packages optimize network routes based on the V and H coordinates and the long haul phone rates applicable. However, AT&T has proposed a new rate structure based on high and low density centers, probably to better meet competition such as Datran and Western Union who also have lines terminating here. Bell has taken the top two levels of its five level switching hierarchy and defined them to be high density centers having lower costs. (The levels are: Regional, sectional, primary, and toll centers and end offices.)[35]

This proposed rate structure may obsolete a number of optimizers on the market today because they do not allow indirect routes through centers where one does not have terminals. New packages have already begun to appear which meet these analysis goals. One is DMW Telecommunications Corporation's "Hi-D Lo-D Network Optimizer." It evaluates direct routings and all indirect routings through high density centers which might be more cost effective than direct routing.

The point of this discussion is that optimizers are available though they may not analyze all possibilities in a network or all combinations of carriers and their rates. Furthermore, these programs can be very cost effective as demonstrated by Dartmouth's saving $100,000 per year after optimizer output suggested a way to multiplex 60 of 300 remote terminals.[36]

Multiplexing saves line costs. The Dartmouth experience demonstrates the effect of using multiplexing to save line costs. However, a recent survey shows that though there are more than 700 firms with communication networks only 7% or 49 use multiplexers to save line costs.[37] Most of the packages mentioned previously analyze line costs but not the cost effectiveness of a particular multiplexing scheme.

James Corliss, Marketing manager of Data Products analyzes the application environments of frequency division multiplexing (FDM) and time division multiplexing (TDM).[38] He says basically, to use TDM with point-to-point systems connected with a high speed channel and FDM with multidrop systems connected with lower speed lines.

Lloyd Bond, President of Timeplex, Inc. agrees that the FDM-TDM question depends on the user cost per channel, but that a 4 channel TDM system from Timeplex can be as cost effective as four separate FMD channels. For example, the four channel Timeplex system will cost $855 or $255 per channel for the common logic and $115 for each channel module or $340 per channel. In contrast every FDM channel is a complete piece of equipment costing about $350. The only time TDM is not feasible is when there is no opportunity to share common logic.[39]

In summary, frequency division multiplexing requires a data set for each channel. Guard bands between channels waste bandwidth, so fewer data channels per voice circuit are provided. But FDM is cost effective for widely dispersed terminals and less costly for fewer channels. TDM on the other hand requires only one data set, which necessitates a module for each channel. But more channels can be provided since guard bands are unnecessary.

Modems. Each terminal requires a modem (modulator/demodulator) to convert the data signal to a form suitable for transmission. To the uninitiated the modem is the transition to the unknown world of telephone lines. To the phone company, when it does not supply the modem, the modem represents the unknown world of data processing. Small wonder everyone blames the modem when communications are interrupted, though it has been established that only 10 to 15 percent of the time the modem is at fault.[40]

When a data circuit hangs up or has intermittent problems the source of error lies either in the terminal, the modem or the line. Independent modem suppliers provide a switch which allows the user to loop the line back to the source, isolating the possible offending modem. If the distant end receives its own signal intact, the user knows the fault lies either in the modem or terminal. The user of an independent modem usually has on hand a supply of circuit boards sufficient to rebuild the modem. Recently, the Bell System has begun to add looping switches to their modems also.[41]

Collect statistics. The best way to avoid data network problems is to keep thorough statistics regarding when, how, what and where errors occur.[42] The statistics should include terminals, modems, circuits, the central computer and software. When the system goes down the time it went down, the time it came up and what the failure was should be recorded. If the carrier is called in, the person who received the call, a service report number and the problem should be recorded. The record of calls and errors is the weak link in maintaining any complex system. Often a look at the record will pinpoint an offending circuit, vendor or piece of equipment which should be replaced.

Also traffic statistics should be collected to prevent the surprises caused by rapid growth. These should include records of system overloads, number of transmissions, line time used, time of day, call destination, etc. This additional information will provide a planning base for future systems. Much of these statistics can be collected by the communications front end. An important reason for keeping this type of traffic statistics is that the Bell System planning statistics are based on the assumption that the busiest hour of the day represents 17.5% of the total day's traffic, which may be the national average but not descriptive of the user's firm.[43]

Data Communication--A Data Administrator Responsibility

Data communication is clearly a responsibility of the Data Administrator, data can be destroyed or degraded within a data communication link just as easily as within the CPU, or more so. The responsibility of the Data Administrator is not one of design but of control of design to ascertain that data accuracy and response requirements are met.

Design control is achieved by assigning the monitoring function to the Data Administrator. The monitoring function collects statistics pertaining to data flow and data channel accuracy considerations which can be used to plan future system changes and monitor ongoing activity. Monitoring techniques and technology which can be used to optimize the design of the data communication system as part of the total computer-based information system will be discussed in the next chapter.

FOOTNOTES

1. "Off-Line Telecontroller Makes 'Swift' Work of 75 Terminals," Computerworld (April 25, 1973), p. 43.

2. "Applications Determine Systems," Computerworld (March 21, 1973), p. 4.

3. J.T. Hootman, "The Computer Network as Marketplace," p. 43 and E. Stefferud, "Management's Role in Networking," Datamation, (April, 1972), p. 40.

4. H. Busignies, "Communication Channels," Scientific American (September, 1972), p. 99.

5. B.W. Stutzman, "Data Communication Control Procedures," Computing Surveys, No. 4, Vol. 4 (December, 1973), p. 201.

6. R.G. Canning, "Future Trends in Data Communications," EDP Analyzer, Vol. 7, No. 4 (April, 1969), p. 7.

7. R.G. Canning, "Distributed Intelligence in Data Communication," EDP Analyzer, Vol. 11, No. 2 (February, 1973), p. 11.

8. Ibid., p. 11.

9. "Minicomputer as Network Front End Signaling New Dispersion of Power," Computerworld (July 26, 1972), p. S12.

10. R.G. Canning, "Distributed Intelligence in Data Communications," EDP Analyzer, Vol. 11, No. 2 (February, 1973), p. 7.

11. J.T.M. Pryke, "A Front-End Primer for IBM Users," Datamation (April, 1973), p. 47.

12. "Smart Users 'Deemphasize' Host," Computerworld (July 26, 1972), p. S10.

13. Pryke, Op. Cit., p. 47.

14. "Smart Users 'Deemphasize' Host," Computerworld (April 26, 1972), p. S13.

15. Ibid., p. S13.

16. "3705 has Limits," Computerworld (July 26, 1972), p. S10.

17. "Smart Users 'Deemphasize' Host," Computerworld (July 26, 1972), p. S10.

18. "Front-End Configurations Based on User's Mainframe Applications," Computerworld (July 26, 1972), p. S14-S15.

19. Ibid., p. S14.

20. Ibid., p. S14.

21. Ibid., p. S15.

22. Ibid., p. S15.

23. Ibid., p. S15.

24. "Front-End Configurations Based on User's Mainframe Applications," Computerworld (July 26, 1972), p. S15.

25. "Smart Users 'Deemphasize' Host," Computerworld (July 26, 1972), p. S10.

26. "Front-End Mini Replaces 270X's, Saves $250,000," Computerworld (April 27, 1973), p. 27.

27. Ibid., p. 31.

28. Ibid., p. 27.

29. Ibid.

30. R.G. Canning, "Distributed Intelligence in Data Communications," EDP Analyzer, Vol. 11, No. 2 (February, 1973), p. 8.

31. Ibid., p. 9.

32. R.G. Canning, "Distributed Intelligence in Data Communications," EDP Analyzer, Vol. 11, No. 2 (February, 1973), p. 1.

33. H.B. Becker, "Information Network Design, Can Be Simplified Step-By-Step," Computer Decisions (October, 1972), p. 14.

34. "New AT&T Rates Could Change Network Designs," Computerworld (April 25, 1973), p. 39.

35. "'Hi-D Lo-D Optimizer' Handles Proposed and Present Rates," Computerworld (April 25, 1973), p. 38.

36. "Multiplexing Costs Based on Volume and Distance," Computerworld (July 26, 1972), p. S11.

37. Ibid., p. 55.

38. Ibid., p. S7.

39. Ibid.

40. "On Choosing an Independent Modem--It All Comes Down to Price, Support," Computerworld (July 26, 1972), p. S2.

41. "Non-Bell Modems Spot Line Faults," Computerworld (July 26, 1972), p. S4.

42. R.A. Kuehn, "Thorough System Records Keep Network Going," Computerworld (April 25, 1973), p. 47.

43. A.L. Katz, "How to Transmit? Phone Company Has Ideas, but . . .," Computerworld (April 25, 1973), p. 35.

CHAPTER VI

MEASURING AND EVALUATING COMPUTER SYSTEM AND DATA BASE EFFECTIVENESS

Introduction

The following chapter discusses the monitoring technology available to the Data Administrator. Monitor results provide the information necessary to control the data flow mechanism of the computer-based information system. Software, CPU's, telecommunication controllers, data base management systems, physical data structure efficiency, data channel error conditions and efficiency of various error detection/correction schemes are all subject to monitor activity. The monitor makes feasible the attainment of a balanced end-to-end system efficiency.

The chapter will first discuss static measures of system module efficiency and their effectiveness; and then proceed to dynamic measurement of an operating system.

Measures of Computer Effectiveness

Comparison of central processor unit add and cycle times. Historically, computer power has been measured by central processor add and cycle times. These figures were used during the period in which most programming was done in machine language. Raw speed was an adequate measure of relative efficiency, since manufacturer supplied software systems were nonexistent and computer architecture did not evince the variety of today. This method is insufficient to analyze the computing systems of today because it ignores (1) the machine architecture, (2) the number of addresses included in an instruction and (3) any consideration of software.

Machine architecture is a significant determinant of machine efficiency. A character addressable machine may perform serial byte manipulation more efficiently than a binary word oriented machine which might perform parallel arithematic operations more efficiently. In a similar manner, word sizes and special features such as hardware instruction look-ahead are ignored by only considering raw speed. Also the data path of a particular machine may be different from that of a comparable system, making cycle times non-comparable.

The comparison of machines having single address instructions with those having multiple address instructions is ineffective since add times reflect instruction fetch times. As an example one might conclude that the IBM System/360 Model 30 with 1.5 microsecond cycle time is faster than the IBM 7094 with 2 microsecond cycle time.[1] This inaccuracy was promulgated by confusing access time with cycle time. Other cases are evident in evaluating total instruction time, as it may require several cycles to perform an addition in a single address instruction machine.

Data must be placed in a register, other data added to it, and the data placed in memory. Thus three instruction set up times (of at least one cycle each) and three execution times (of at least one cycle each) are required to perform the addition. While in a multiple address instruction all steps may possibly be performed in one cycle time, but at the cost of longer, more expensive instruction words and a more complex hardware decode logic.

A final failing of this method of machine comparison is that no consideration is given to the efficiency of manufacturer supplied compilers, assemblers or operating systems, nor the particular job mix the machine will face. However, the most important failing of this method of performance analysis is that "no one instruction is typical of the operation of an automatic digital computer."[2] Furthermore, the predominance of nonarithmetic instruction executions such as loads, stores, branches and input/output is not considered by this technique.

Instruction Mixes

As an attempt to circumvent the major criticism of using add times to compare machines, the concept of an instruction mix was developed. The concept derives from the fact that if two computing systems are compared with respect to a number of instructions, the ratios of instruction execution times will differ for different instructions. In the method each instruction in the repertoire of a computer is assigned a weighting factor derived from empirical frequency distributions of instructions which may represent different jobs, classes of applications or even an attempt to characterize the entire general area of scientific or commercial data processing. Applying the weight to each instruction provides an average instruction time that can form a basis for comparison between two or more machines. And because the rating is derived from the job environment, machine load is implicitly considered.

Instruction mixes are often popular because they are easy to use. Detailed instruction times are available and it is relatively easy to gather statistics on the frequency of use of each instruction. The gathering of statistics, however, is also a weak point in the analysis. These statistics are usually gathered by dynamically tracing the execution of several programs through a large number of instructions. An important problem is the determination of how many programs must be traced and how many instruction executions constitute a reasonable size program. Other difficulties arise, such as subjective decisions as to how frequently the field size may cross word boundaries in fixed-word systems, or what mix of instructions will be included in the miscellaneous category which is itself an instruction mix.

Moreover, the composition of a mix itself is dependent upon the structure of the CPU from which the mix was originally derived. It therefore becomes less and less applicable as the structure of the CPU under study departs from the original structure. As an example, a system having only one accumulator and one multiply register cannot be compared to a system having multiple accumulators. The use of load and store instructions for the latter configuration will be less. Other considerations are:

(1) Single-address versus multiple-address logic
(2) Word sizes
(3) Fixed length versus variable length systems
(4) Index registers versus only indirect addressing
(5) Effect of data-base conversion time[3]

Finally, the power of a machine's instruction repertoire is neglected. The method does not consider the possibility of using different numbers of instructions to perform the same task. As an example, a group of experienced system engineers were asked to specify the time in microseconds required to execute a mix of instructions. The mix was based on an IBM 7090, but the predictions were to be of the time required on a System/360 Model 40. The answers ranged from 11.88 to 30.66 with a mean of 21.5 and a standard deviation of 7.0.[4] Therefore, comparisons of machines having different instructions is self defeating.

In summary, items such as differing instruction sets, word size, data path, addressing logic, special features and software are not considered. Also the various mixes do not include any I/O considerations, which are very important in business and many scientific applications.

Therefore, although instruction mixes represent some improvement over simple cycle and add times, they are still inadequate for hardware comparisons and completely inapplicable for software evaluation.

Kernel Programs

A much better technique to appraise basic internal performance is kernel problem comparisons which circumvent the difficulties posed in comparing systems having different addressing capabilities. A kernel program is a typical program which has been partially or completely coded and timed. Thus it derives its name from the fact that the essential part or kernel of the application under study is examined. The timings are based on the manufacturer's stated execution times for the instructions that comprise the kernel program for a given machine. Both simple algorithms such as matrix inversion and quite elaborate data processing applications such as payroll systems have been coded and timed as kernel programs. The Auerbach EDP Standards includes a compendium of such programs.

Kernel problems may take the form of (1) polynomial evaluations which measure indexing speeds as well as arithmetic speed, (2) matrix operations, (3) evaluation of a particular formula which hopefully contains sequences typical of arithmetic operations, (4) miscellaneous routines to include, bit manipulation, interpolation, random number generation, (5) comparison of detail transactions with master records and sequence-checking of both files, (6) internal processing for a particular activity, (7) formatting a typical line in a printed report, and (8) routines such as table look-up, block transfers, etc.

Kernels overcome some major limitations of instruction mixes because they are machine independent and permit the use of a system's complete instruction set and special features such as addressing logic, use of special registers, indexing ability, etc. These advantages are also limitations in that so much is dependent upon the programmer's expertise and honesty. It is not unusual for a single programmer to be better at programming one machine than another or for one programmer to be better than another in terms of the efficiency of the code he generates. The problem, therefore, is how to account for this bias in comparing a set of kernels across a variety of machines. The answer is that there is just no way to explicitly and accurately account for this bias other than to assume it is negligible and trust the honesty, integrity and expertise of those who generate the kernels.

Another problem akin to weighting the effect of instructions in a mix is the proper weighting of several kernels which are used to define a system environment. Multics, for example, found that programmer's intuition was inaccurate in predicting which modules were most commonly used in the system.[5] Other difficulties arise because (1) the wide range of I/O instructions used in practice is not considered, (2) factors in evaluating systems with different organizations may be overlooked, such as the effect of BCD to binary conversion, and field size relationships to word boundaries, (3) significant variations may exist between the relative internal performance of kernels across machines, and (4) kernels generally reflect hard core processing and ignore the effect of the many administrative and bookkeeping type functions which must be performed by machines.

In most cases, kernels reflect a desk calculation of internal power. However, they are only applicable to comparisons of compatible families of machines as they do not consider a broad enough integrated problem environment to accurately reflect differences in widely differing computing systems. The relative power of a system is not necessarily how fast it is internally, but how fast it can perform the complete job. The interaction of internal performance with I/O speeds and facilities as well as programming systems efficiency must be considered when evaluating a total system.

Henry Lucas suggests other uses for kernels.[6] For example, the quality of compiler output code can be judged by programming and compiling a kernel and then examining the output code. However, they are inadequate to evaluate the quality of vendor supplied software and are inadequate for the evaluation of either present or projected software performance.

Benchmarks

A benchmark is an existing program that is coded in a specific language and executed on the machine being evaluated. It is the only reliable and accurate measure of system performance as long as the programs selected are run in the large and small mixes expected in the actual operation, thus providing a good indication of the large and small problem behavior of the system.

A comprehensive series of benchmark runs can demonstrate differences in machine organization and evaluate I/O performance as well as secondary storage. The benchmark is the first evaluation technique to include a consideration of software, as speed of compiler and execution are included in the measure. For example, the same program may be coded in assembly language or a higher level language and the speed of each compared. Also a series of benchmarks may be run under control of an operating system and then under multiprogramming to demonstrate the increased throughput.

The limitation of benchmarks is that they may be used too often by eager salesmen to substantiate claims. The truth is that it takes a large effort to program benchmarks and to prepare realistic data. Benchmark performance may also be misleading in that the figures are not usually obtained under a capacity load and that extrapolation to programs not benchmarked may be a risky business.

In consideration of software it is important to note whether the software is designed to execute rapidly but compile slowly, etc. Furthermore, one must note that software improves over time. Such considerations must be used by the analyst to modify the results obtained by benchmarking.

The job mix used for benchmarking should also be a reflection of the types of programs actually run on the system. Compute bound and I/O bound programs, disk and tape routines and a priority of runs must be considered in proportion to the real job mix. Such specification is almost impossible due to the varying loads seen by a system throughout a day and over longer periods of time. In effect no one can specify what exactly the load looks like, therefore, benchmarks are of limited use. One exception is as a before-and-after test to monitor performance following system change.

Thus the basic assumptions of benchmarks is that the present job mix is an adequate predictor of future activities. It is the first measure of effectiveness which measures software-hardware interaction. But it fails in that it is difficult to run sufficient benchmark programs to insure that the job stream has been adequately represented.

It is possible to use a software monitor to gather statistics on most typical jobs but the above mentioned constraints become even more severe as a multiprogramming environment is entered. It takes a large number of jobs to fill the queues and get past the start-up stages of a multiprogramming environment.

Synthetic Programs

Synthetic programs are used where the job stream is less certain or where standardized tasks exist. It is coded and executed like a benchmark, but like a kernel it is not representative of any existing piece of code. However, unlike a kernel, it is coded and includes considerations of I/O, files, and the environment provided by the operating system.

The activities included in a synthetic program cover a wide range. Some program segments are compute bound and others make high demands on I/O. Bucholtz suggests that the programs be highly cyclic so that I/O to compute time ratio comparisons may be made between systems or between operating in sequential or multiprogramming mode.[7]

A major advantage of the synthetic program is that it offers the most flexibility of any measure of effectiveness yet considered since any type of job stream can be modeled, i.e. jobs can be designed to include any desired measurement parameter. However, many of the problems of benchmarks also exist with synthetic programs, such as the representation of the current or projected job mix, job priority, job selection, the weighting of different synthetic programs, and the adequate attention to the effects of the programming system. Applying synthetic programs to time-sharing systems would be an immense problem just to mobilize the manpower to code and run the programs. In retrospect synthetic programs are doomed to failure because inadequate standards would reduce their value for making comparisons among systems. Their attribute of flexibility is also their inherent disadvantage.

Another disadvantage is that a large number of synthetic modules must be used to adequately represent the system, however, no standard synthetic modules are required. The same problem pertaining to the large number of modules required to start-up a multiprogramming system holds for synthetic programs as it does for benchmarks. Similar difficulties also arise in forecasting an anticipated job mix. However, synthetic programs are useful in a before-after test when performance monitoring indicates the need for hardware changes, or if proposed changes exist as synthetic modules they can be tried on the appropriate hardware.

Modeling Computer Systems

<u>Simulation models</u>. Simulation is the most powerful and flexible of the evaluation techniques as it provides a testing ground for and insight into the functioning of the system.[8] Simulation involves building information models of the system structures and then exercising these models with assumed models of programs and data. Two basic types of simulation have been used to evaluate computer performance--the event-oriented unit step model and the empirically derived model which uses actual data distributions to represent the specific configuration and job load.

The generalized simulation model to describe any computing system does not exist. Computers differ so much in their organization that each simulation must be custom tailored to a certain class of machines. One commercial simulation tool is SCERT, a program product of COMPRESS, Inc. SCERT appears to provide good results for serial system architecture when the programs to be run are carefully formulated for each computer studied. It has not yet demonstrated to be sufficiently accurate in studying system architectures which employ dynamic resource allocation for concurrent processing, such as used in the B5500.[9] The processing of the model consists of a series of table lookups, and empirical equations are used to estimate the behavior of the simulated job mix. Though the system is not a clock-oriented event simulator, the system does attempt to assess the degree of concurrency and to include real-time factors using a clock oriented event simulator.

Event-oriented simulators have been directed towards a single system. Other simulators have been designed to simulate a family of machines. And one developed by Lockheed (LOMUSS) is a more general purpose language which views the computer facility as a job shop or PERT network.

The disadvantages and problems of a simulation of computer systems are similar to those of simulation in general. There is always the question of validity and proving a simulation correct. The results are no better than the assumptions of the model. If the model is too detailed it is costly and extremely system dependent, if the model is too broad it yields poor information. Furthermore, when used as an adjunct to system design, often the actual system is designed and operational before the model is complete.[10]

The major advantage, however, of constructing a simulation model is the insights into the system operation it provides. The act of modeling in itself is inherently good. The danger of simulation is placing too much confidence in its outcome. The results are very sensitive to the assumptions used to approximate the model and the data used to drive the model. Often small omissions or approximations can produce large discrepancies between performance of the model and that of the real system. However, it is in projecting the performance of proposed systems that validation problems are most severe. Simulation is excellent for performance monitoring as it provides a model of the system on which to try changes.

Analytic Models

An analytical model mathematically describes the performance of a computer system. Analytic models are normative in that they are used to study the optimal characteristics of a system. Their construction and operation is thought to be more cost effective than simulation models which require huge amounts of machine time and set-up time. However, analytic models do not generally include a comprehensive set of operating system functions, nor do they consider the quality of software performance. Furthermore, it is difficult to include the random effects of multiprogramming and multiprocessing in an analytic model. In many cases the entire system is too complex for an analytic model, especially when the interactions between hardware, software, applications programs, and a sophisticated interrupt structure must be considered. Often the simplifying assumptions necessary to develop large analytic models reduce their validity.

However, description of a simplistic, uniprocessing subsystem is the forte of an analytic model. These models may be included as integral parts of large system simulations. Many of the currently available analytic models are based on Markov chains. Such models normally describe the performance of some subsystem, such as a CPU scheduling algorithm, and I/O model, a memory management model, and a disk seek model. Other modeling techniques which might be applicable include optimal control theory and mathematical programming. It must be remembered that the goal of modeling is not to construct a faithful representation of the real world but rather to obtain insight for designing the system and to determine potential causal relationships between input and output. Analytic models are best used as components of simulation models.

Monitoring

Introduction. Monitoring is the collection of statistics and actual performance parameters of an operating, live system. Gary Carlson of Brigham Young University identifies three goals of monitoring: (1) save money, (2) understand the system better and (3) indicate future developments and trends that will lead to better service and lower costs.[11] Monitoring then is applied as a means of obtaining a measure of system efficiency, much as a time-study engineer would monitor an entire operation in a manufacturing plant and the components of the operation. Thus monitoring may take a macro view of the operation which would concentrate on discovering problem areas or monitoring may take a micro view which would concentrate on discovering the specific cause of a diagnosed difficulty.

Performance monitoring studies thus range from feasibility studies through post-implementation hardware configuration. Techniques are used basically to maximize throughput, decrease turnaround time, identify levels of utilization, modify device allocation and determine modification requirements.[12] Other uses of the data collected from monitoring a live system are: (1) determine the job load in uncertain environments; (2) system tuning; (3) pointing out bottlenecks in operations; (4) improve I/O overlap, I/O device analysis, and balancing of partitions; (6) investigate program behavior prior to implementation; and (7) the collection and analysis of statistics to determine user profiles for policy decisions.

Usually the measurement novice homes in on monitoring central processor, channel, controller and device activity. Examination of this type of measured data can lead to elimination of device and channel contention and lead, through reconfiguration, to increased throughput for a better balanced system. While reconfiguration analysis is valuable and should be included in any measurement program, greater payoff may often be achieved by monitoring data base activity, problem programs and the operating system itself. A good monitoring session should produce sufficient data to direct optimum data file reorganization and by pinpointing frequently run programs and subroutine modules reveal the areas in which optimization of code can significantly reduce execution time and throughput.

In second generation machines in which one job is performed at a time, elapsed time on serial tasks was the only serious consideration in determining throughput and job time and in comparing workload and available machine power in a specific set of applications. In third generation machines, with multiprogramming, multiprocessing, and virtual memory systems this simple measure is no longer a good indication of the efficiency of the system. In many installations only 25 percent to 35 percent of the CPU cycles are utilized.[13] The indication being that the same workload may be processed on a smaller machine or that an increased workload may be processed on the same machine if the system is properly "tuned."[14] Tuning a system requires that new measurement devices, measurement techniques and new parameters to measure be found which better reflect the true system efficiency.

Computermetrics. The biggest problem facing developers of measurement systems is that no one knows exactly what to measure. Measurement and performance evaluation of computer systems is "an art trying hard to become a science."[15] Everyone has his own ideas of exactly what parameters should be measured to gain a handle on the effectiveness of a complete computer system.

The art of measurement is now turning into a science. During a keynote address at Computer Science and Statistics: Fourth Annual Symposium on the Interface in September 1970, Hamming proposed the term compumetrics for measurement of computer systems. Hamming stated:

> The director of a computer center is responsible for managing the utilization of large amounts of money, people and resources. Although he has a complex and important statistical problem, his decisions are normally based upon the simplest collection and analysis of data--since he usually knows little statistics beyond such elementary concepts as the mean and variance. His need for statistics involves both the operational performance of his hardware and software, and the environment provided by his organization and users.
> A new discipline that seeks to answer these questions--and that might be called 'compumetrics' --is in the process of evolving. Karl Pearson and R.H. Fischer established themselves by developing novel statistical solutions to significant problems of their time. Compumetrics may well provide contemporary statisticians with many such opportunities.[16]

During April 1971 a Special Interest Committee on Measurement of Computer Systems, SICMETRICS, was formed within the Los Angeles Chapter of the Association for Computing Machinery. This was followed by the ACM Special Interest Group on Operating Systems, SIGOPS, Workshop on System Performance Evaluation in April 1971. During November 1971, the ACM Special Interest Committee on Measurement and Evaluation, SICME, was formed being elevated to an ACM Special Interest Group on December 8, 1972.[17] Since then numerous organizations have had special sessions on measurement and evaluation of computer systems.

Evaluation technique. The process of evaluating computing systems involves the use of new tools, such as hardware monitors, software monitors and hybrid monitors as well as the application of new techniques to old tools such as accounting data. Dr. Gary Carlson, Director of the Brigham Young University Computer Center, thinks that monitoring requires a pragmatic approach since little theory is yet available.[18] Bell, Boehm and Watson extend this notion in enumerating seven steps or phases of systems improvement analysis:[19]

(1) Understand the whole system in terms of management organization, hardware and software utilized, the system load, the parameter being monitored and the data being collected.

(2) Analyze the operations by collecting more detailed and quantitative data than collected previously. Bottlenecks and the criticality of several inefficiencies should be revealed.

(3) Formulate a performance improvement hypothesis based upon the previous analysis.

(4) Analyze the cost effectiveness of these possible improvement modifications.

(5) Test the various hypotheses by performance monitoring and other appropriate measurement techniques.

(6) Based on the results of the tests, implement the appropriate combination of modifications, being careful that two changes do not cancel one another.

(7) Test the effectiveness of the modifications. Reiterate phases three through seven if necessary.

Monitoring difficulties. The central problem of the monitor analyst is the development of a monitor model of the system being analyzed. From the particular model developed, several hypotheses evolve. Tests of these hypotheses hopefully reveal what changes might be made to increase effectiveness. Without a coherent yet simple model of the system one too easily falls into the trap of problem finding through measurement. This is a hopeless task due to the thousands of possible variables which can be monitored and the fantastic speeds at which data can be generated by a machine performing a billion operations per second. It is dangerously easy to collect massive amounts of confusing data unless one establishes an experiment with a clear hypothesis in mind.

Once an hypothesis has been developed and the proper variables to be measured selected, questions arise as to (1) whether or not the intended variable was measured and (2) if the proper variable was measured, does measurement during this particular time segment accurately reflect system performance? It is too easy to connect a hardware monitor's probe to the wrong point or make logical errors in combining two or more probe signals. Software monitors must be reprogrammed for each new measurement and offer no less an impediment to accuracy and validity. Carlson suggests that two confirming measurements always be used and that consideration always be given to other possible interpretations of the monitor information.[20]

The important point is that one should not rush headlong into the collection of data without first having developed a testable hypothesis.[21] It is easy to collect massive amounts of confusing data unless a monitoring experiment with a clear, testable hypothesis is established.[22] In fact Bell indicates that modeling data collection mode, experimental design, and data analysis are even more difficult and important than choosing data collection tools.[23] Furthermore, one should realize that monitoring can only indicate the present limitation on performance, but it provides no information on how much improvement can be made before another factor becomes limiting.[24]

Monitoring By Observation

Computer system monitoring often requires no sophisticated hardware and software monitors but only the eyes and ears of the performance measurement analyst. For example, the sound equipment evokes may be indicative of problem areas. A multiprogrammed system may be experiencing severe disk contention in attempting to print spooled output thus producing strongly synchronized printing from the line printers in a large installation. This occurs when printer A must wait a fraction of time while the disk head moves to another track to dump some output onto printer B. When one printer completes output for its job, the other printers begin operating at a sharply increased rate.[25]

One difficulty with generalizing sound to activity was noted by Carlson.[26] Almost always when the CPU activity is monitored and its activity is below eighty percent, say in the thirty to forty percent category, the operational people explain that the measurement was taken at mid-month and that CPU activity would increase if measured at month-end. However, in several cases when the analysts have returned to measure their most active time, almost without exception the net effect of "peak loads" was a slight increase of two percent to four percent CPU active, and a large relative increase in printer busy. Printer activity typically runs eight to twenty-five percent with peak load activity at ten to forty percent. This gives the operators more paper to handle and more noise and apparent activity.

Another useful tool is the incandescent console light. For example on the IBM 360's, the Wait and System lights are the most useful. The Wait light indicates CPU idle and the System light indicates system activity. Because the lights are incandescent they crudely integrate system activity over time. A bright wait light, for example, may indicate an I/O bound system. Other lights on channels and controllers indicate activity on those resources.

However, two cautions are in order. Lights indicate very gross measurements, for example, Bell indicates that one could distinguish the difference between 35 percent activity and 70 percent activity but not between 70 percent activity and 100 percent activity.[27] Another difficulty resides in the fact that an indicator may not in fact measure what it is purported to measure. For example, a measurement group from Mobil Oil reports that the IBM "CPU meter" is a misnomer, "system meter" being more descriptive as the meter continues if any device in the system is running.[28] A tape drive in the rewind state or a printer in the ready state will all cause the meter to run.

Other important data can be obtained simply by watching or logging operator activity with a view to increasing their efficiency through proper equipment layout and job scheduling. For example, a bank replaced its IBM 7074 computers with a 360/50 and replaced that with a 360/65 but got no higher throughput. By watching the CPU wait light the bank personnel decided that seven jobs consumed all available CPU time. After several days of monitoring it was determined that the seven jobs were indeed CPU bound but only took three percent of the total active CPU time. Since the system was active fifteen percent of the total system time, the problem was system logistics. Many short jobs required new I/O setup, and the physical system was crowded and disorganized. To further confound matters the bank used eighty character records which induced extreme rotational and seek-time delays. When the physical problems were solved, the work could have been performed on a

360/40.[29]

It is important to utilize all available information in constructing an effectiveness model of the computer system, but one must appreciate that visual and audio data are only indicators of macro problems which should be investigated through the use of hardware and software monitors.

Software Monitors

A software monitor is a program which collects system data by interrupting the operating software system and querying the contents of various registers and memory locations. The simplest example of a software monitor is a job accounting routine. More sophisticated systems collect statistics which may reflect core usage, queue length or individual program operations.

Advantages. A distinct advantage to this type of monitor is that it has access to all tables and all registers within the machines. This allows system activity statistics collected to be precisely correlated to system load. Also the monitor is to a degree intelligent in that it can test various parameters to decide how much or how little data to collect. Since it is software, it provides a wide latitide of flexibility in selecting parameters or making changes. Furthermore, it is relatively easy to learn to use and its functioning is apparent to neither the system operator or the computer system maintenance crew.

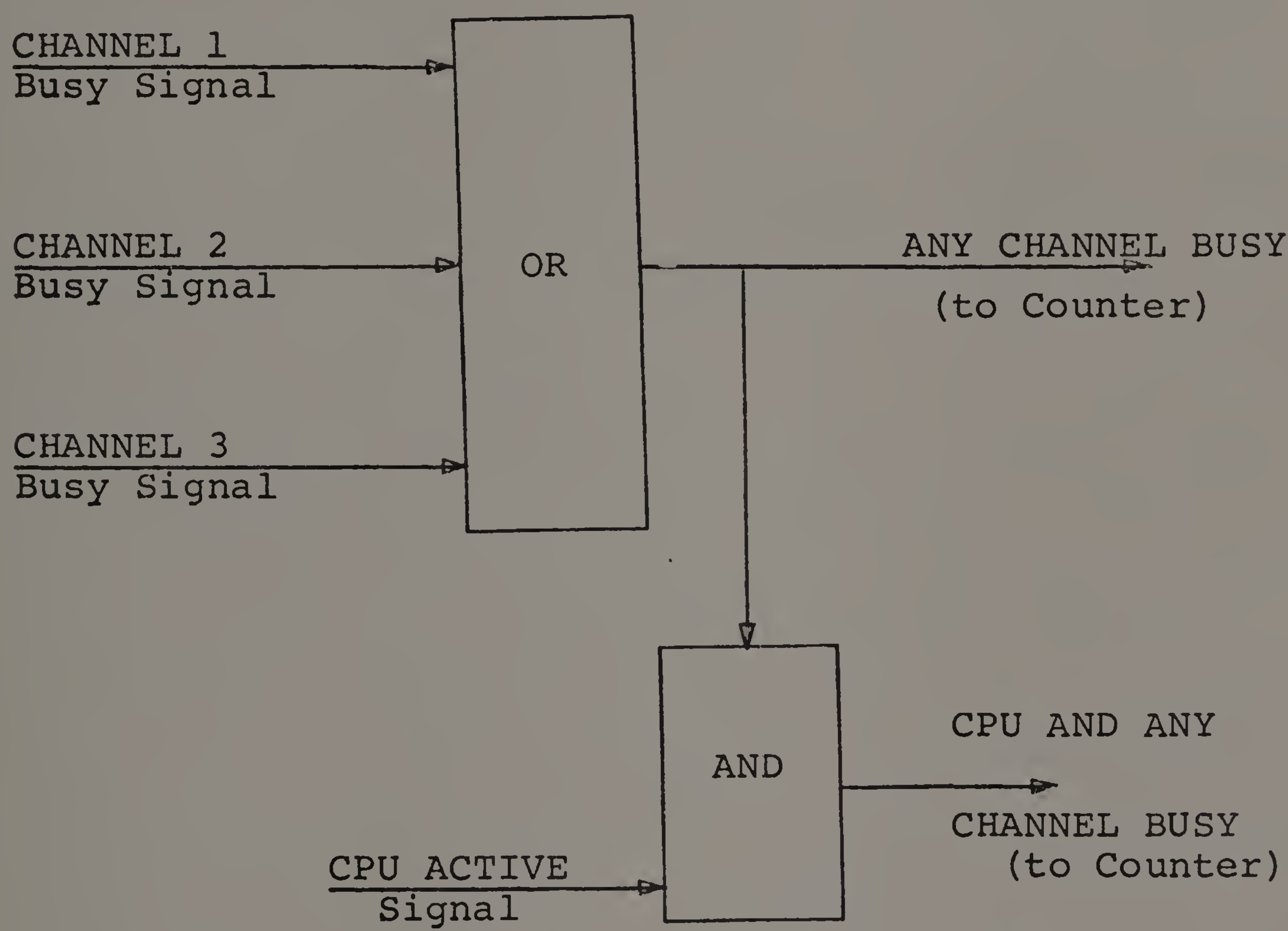
Disadvantages. The central advantage of software monitors is their flexibility and simplicity of operation. Their prime disadvantage is that a new monitor is required for each hardware and operating system configuration. Often the operating system must be changed to collect data. In some cases the operating system may mask the monitor program interrupt causing the completion of an input/output operation not to be recorded until long after it occurs.

The total system efficiency is degraded by measurement. Since the data gathering routine is a program, it uses memory space, CPU cycles and channel time. Thus the process of measurement affects the values of the parameters sensed. In some instances the monitor may require an entire channel, resulting in the measurement of a degraded system. Also there is no access to the actual hardware activity at channel end since monitoring can be performed only within the CPU and a channel register need not reflect the true activity being performed at the distant end. Other difficulties arise due to the nature of computers and the granularity of the accessible software timer. Since computers operate sequentially, simultaneous measurements may not be taken and the precision of these measurements is dependent on the precision of the accessible timer which is often 1/60 second.

Hardware Monitors

A hardware monitor is a device which collects signals from a host computer through high impedance probes attached directly to the computer's circuitry. The signals usually are passed through a Boolean plugboard where wired logic can produce data on combined functions such as total I/O time or I/O and CPU overlap or any more complex relationship may be monitored. For example, in a three channel system signals for each channel would normally be passed through an OR gate and into a counter to provide a channel busy indicator. If this composite signal is mixed in an AND gate with the CPU active signal the resulting signal will indicate CPU and I/O overlap. (See Figure 1). The sensed information can then be transmitted to a counter and periodically stored on magnetic tape for later analysis.

Advantages. Since hardware monitors can sense any arbitrary signal they may be used on any computer irrespective of manufacturer or hardware used. No new measurement tool is required for each system. Other advantages of hardware monitors is that their use does not contaminate or degrade the system being measured. They require no overhead, no memory and no CPU cycles. They can be activated and deactivated without affecting processing rates. Furthermore, system catastrophies do not terminate data collection and operator errors are particularly detectable. Direct measurement on peripheral devices and simultaneous measurement of several inputs is possible.



HARDWARE MONITOR BOOLEAN PLUGBOARD SCHEMATIC

Figure 1

<u>Disadvantages</u>. Disadvantages of hardware monitors center around the large mass of wires and probes often needed to affect measurement and the substantial learning time required (about six months).[30] The wires and probes are inconvenient to operators and maintenance personnel. The large number of probes and possible probe points leads

to a definite possibility of error no matter how careful the analyst.[31] Also extensive documentation must be listed for each computer system enumerating the possible probe points. Resolution of the counters and the sensitivity of the probes used limits the parameters which can be measured and the speed of the devices which can be monitored. Richard Rudd of Applied Computer Technology states that probes and counters are not fast enough for the super-sophisticated computers on the market.[32]

Finally, once data is recorded it is extremely difficult to relate the data to the actual job load. There is also no mechanism to select probe point values. Data is generated rapidly and non-selectively. Sampling, to some degree aleviates the problem by avoiding synchronization with the machine. However, operating a hardware monitor is still akin to measuring a black box, i.e. a system whose internal workings are unknown. Thus, often a hardware monitor can center attention on a problem area but can do nothing to identify or resolve the conditions causing the problem.

Hybrid Monitors

There is a current trend towards integrated monitoring in which software and hardware monitors are combined in a minicomputer controlled monitoring system which utilizes hardware monitors to best advantage in their ability to monitor activity without disturbing performance and the software monitor to deliver to the minicomputer register contents of the measured system so that hardware monitor results can be correlated to system load. Compress has developed the Dynaprobe 8000 to this end. The monitor uses a minicomputer to determine which connected probes to monitor and when to record their values. Additionally, the mini can interpret data from the host computer to determine which programs are active, which pages or overlays are in memory, etc.

Future hybrid monitors will combine hardware and software monitors with the job accounting system to provide complete data for performance analysis. Conversational monitors for real-time information based on a user's questions about system performance will also be available, with the on-line program probably residing in a minicomputer.[33]

Organizing For Performance Measurement

State Farm Insurance decided to shift the workload of its more or less independent regional data processing centers to a centralized site. The shift involved a changeover from batch oriented programs written in Autocoder and run on 360/30's and 40's in emulation mode, to pseudo real-time PL/1 programs to be run on 370/165's with 360/30's used as intelligent terminals in regional offices.

Faced with such a drastic change, State Farm began to explore all sorts of measurement tools--hardware and software monitors, simulators and program performance evaluators. A study indicated the company might save .8 percent of data processing expenses per analyst per tool used.

Following a well structured strategy that moved from original definition of objectives, through a feasibility study and reports to management, into development and implementation of a measurement group, the company now has four types of tools and eight analysts divided into two teams.

Each of the team members has become strong in the use of a particular tool, but project leadership rotates among them to enhance all their skills. Normally, one plans the project, another installs the proposed changes and a third evaluates the results. The fourth member adds his expertise where it will do the most good.

The technical support group, including the measurement teams, has the same status as programming and operations, and leaders of each group report to the same vice-president. There is therefore no built-in clout due to organizational position, and recommendations have to be backed by strong evidence before they are considered.[34]

Mayford L. Roark, Director of Systems at The Ford Motor Company reported on results of tuning 3 data-base management systems.[35] In case A, running time was reduced 80% after system tuning; in case B, running time was reduced 83% after system tuning; but in case C, run time increased by 1000% when a DBMS was implemented, and acceptable efficiency was achieved only after months of tuning. As systems become more complex the necessity for monitoring to achieve increases in efficiency increases.

Data Administrator Monitoring Responsibility

The corporate information system includes the data input system, the telecommunications system and the data management system, which taken together are extremely complex. Data flow through and in these systems must be controlled to insure its accuracy, integrity and security and to insure that system response requirements are met. The tool used to control data is the integrated hardware/software system monitor. If the Data Administrator is to be given data responsibility, he must be given a means to control data--direct responsibility for the monitoring function.

Returning to our model of the Data Administrator as an Aerospace System Design Engineer we see that in the initial design the System Engineer uses vendor data, his own test results and analytic and simulation models to select sensors and control design of the system. This is akin to the use of computer and peripheral vendor data and benchmarks, analytic and simulation models by the Data Administrator. Once the prototype is constructed, the System Engineer uses instrumentation (voltmeters, ammeters, oscilloscopes, computer-based monitors, etc.) to examine and analyze the workings of the system. As the design evolves because of modification or the requirement to meet a changed control environment, additional measurements are taken and the design dynamically changed to meet the new conditions. In a similar manner the Data Administrator uses the instrumention of the hardware/software monitor to examine and analyze the workings of the mechanized data flow structure of the organization (computer-based information system). Monitoring is a continuing activity made necessary by the dynamic software/hardware system and the dynamic control environment of the organization.

The monitoring function is the last element in the Data Administrator Model. The concluding chapter integrates all the foregoing chapters into a single Data Administrator Model, summarizes important points discussed earlier and suggests new research directions.

FOOTNOTES

1. P. Calingaert, "System Performance Evaluation: Survey and Appraisal," Communications of the ACM, Volume 10, Number 1 (January 1967), p. 14.

2. Ibid., p. 14.

3. R.A. Arbuckle, "Computer Analysis and Thruput Evaluation," Computers and Automation (January 1966), p. 13.

4. Calingaert, Op. Cit., p. 14.

5. H.C. Lucas, Jr., "Performance Evaluation and the Management of Information Services," Data Base, Vol. 4, No. 1,(Spring 1972), p. 4.

6. H.C. Lucas, Jr., "Performance Evaluation and Monitoring," Computing Surveys, Vol. 3, No. 3 (September 1971), p. 83.

7. W. Bucholz, "A Synthetic Job for Measuring System Performance," IBM Systems Journal, Vol. 8, No. 4 (1969), p. 314.

8. P. Calingaert, Op. Cit., p. 14.

9. R.R. Johnson, "Needed: A Measure for Measure," Datamation, Vol. 16, No. 17 (December 15, 1970), p. 24.

10. D.J. Campbell and W.J. Heffner, "Measurement and Analysis of Large Operating Systems During System Development," AFIPS Conference Proceedings, FJCC, Vol. 33, Part 1 (Montvale, New Jersey: AFIPS Press, 1968), p. 910.

11. G. Carlson, "How to Save Money with Computer Monitoring," Proceedings of the ACM Annual Conference (Boston, Mass., 1973), p. 1018.

12. Seymour Jeffery, Chief of the Systems Development Division at the Institute for Computer Sciences and Technology of the National Bureau of Standards, reported in E. Drake Lundell, Jr., "What System Parameters Should Be Measured?," Computerworld, December 13, 1972, p. 9.

13. R.G. Canning, "Savings from Performance Monitoring," EDP Analyzer, Vol. 10, No. 9 (September 1972), p. 1.

14. Ibid., p. 4.

15. Dr. Arnold F. Goodman, McDonnell-Douglas, speaking at the 1973 FJCC on "Measurement of Computer Systems: Executive Viewpoints," reported in "Measurement: Art or Science?," Computerworld, (December 1 , 1972), p. 9.

16. A.F. Goodman, editor, Computer Science and Statistics: Fourth Annual Symposium on the Interface--An Interpretive Summary, (North Hollywood, Calif.: Western Periodicals Company, 1971).

17. Chairman's Column, Performance Evaluation Review, Vol. 1, No. 4, p. 2.

18. Carlson, Op. Cit., p. 1018.

19. T.E. Bell, B.W. Boehm and R.W. Watson, Computer Performance Analysis: Framework and Initial Phases for A Performance Improvement Effort, The Rand Corporation, R-549-PR, August 1971, pp. 6-11.

20. Carlson, Op. Cit., p. 1022.

21. Canning, Op. Cit., p. 9.

22. R.R. Johnson, "Needed: A Measure for Measure," Datamation, Vol. 16, No. 17 (December 15, 1970), p. 25.

23. T.E. Bell, "Choose Your Tools to Check Your Computer," Computer Decisions (November 1972), p. 15.

24. H.C. Lucas, Jr., "Performance Evaluation and Monitoring," Computing Surveys, Vol. 3, No. 3 (September 1971), p. 86.

25. T.E. Bell, Computer Performance Analysis: Measurement Objectives and Tools, The Rand Corporation, R-584-NASA/PR, February 1971, p. 11.

26. Carlson, Op. Cit., p. 1021.

27. T.E. Bell, "Choose Your Tools to Check Your Computer," Computer Decisions (November 1972), p. 13.

28. M.D. Sewald, M.E. Rauch, L. Rodiek and L. Wertz, "A Pragmatic Approach to Systems Measurement," Computer Decisions (July 1971), p. 40.

29. C.D. Warner, "Monitoring: A Key to Cost Efficiency," Datamation, Vol. 17, No. 1 (January 1, 1971), p. 42.

30. Canning, Op. Cit., p. 14.

31. "Users Best Catalyst for New Tools," Computerworld, December 13, 1972, p. 9.

32. Ibid.

33. Seymour Jeffery, Chief of the Systems Development Division at the Institute for Computer Sciences and Technology of the National Bureau of Standards, reported in E. Drake Lundell, Jr., "What System Parameters Should Be Measured?," Computerworld, December 13, 1972, p. 12.

34. D. Leavitt, "System Evaluation Must Be Planned," Computerworld, January 10, 1973, p. 8.

35. M.L. Roark, "The Hazard of Getting Trapped," presented at "Managing the Impact of Generalized Data Bases," a special session at the National Computer Conference 1973, New York, session not included in Proceedings, but published by the Special Interest Group in Business Data Processing, the Association for Computing Machinery.

CHAPTER VII

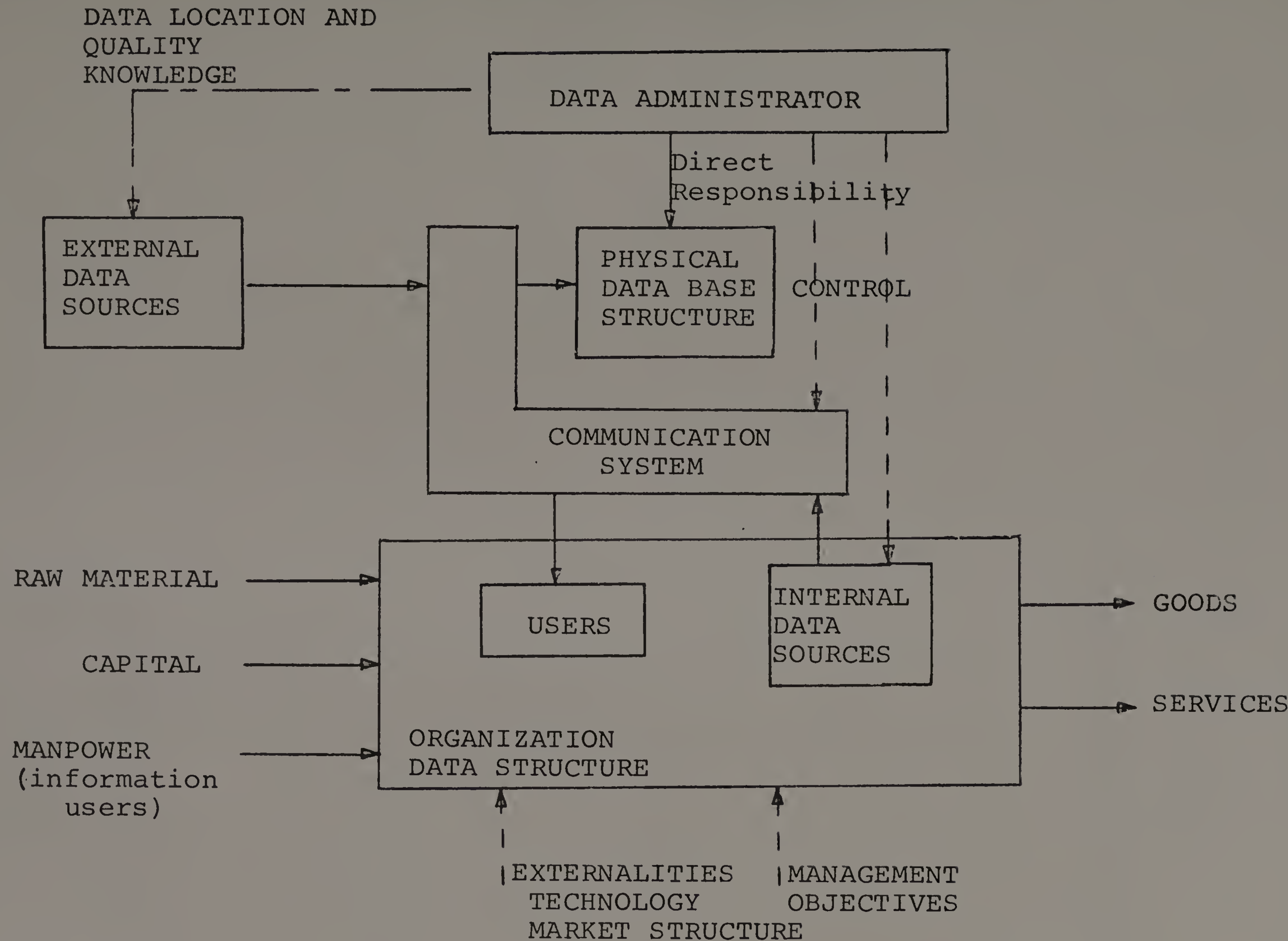
CONCLUDING REMARKS

The Data Administrator Model

A Data Administrator Model appears as Figure 1. Input raw material, capital, and manpower (data users) is passed through the organization data structure, constrained by technological and market structure externalities and management objectives. The output is goods and services. The organization data structure is a network of internal data sources and data users. A web of communication connects all the data users, and internal and external sources to the data base.

The Data Administrator has direct responsibility for the physical data base and its structure, in other areas his data responsibility is expressed in operational control of internal data sources and the communication system. External data source location and quality knowledge allow selection of data appropriate to user applications.

Data control and direct data responsibility implies regulation of data accuracy/integrity quality and data security in the sources, uses and data base processing of data. The data function has always existed in organizations, but has been part of the general management function. What has happened is that the data base concept has forced data sharing among disparate users, the result is the con-



DATA ADMINISTRATOR MODEL

Figure 1

centration of the data function in the Data Administrator and the recognition of data as a corporate resource to be managed, like money or other scarce resource.[1]

This concentration of data responsibility grew out of the need for someone to arbitrate data among the various users and because of the high degree of technology involved in managing data in a computer-based information system. Data management implies the planning, organizing, coordination and control of data resources so that individual users get the right data within the right time constraints to maintain control over organizational processes.

This is the data system control criteria of the aerospace system design engineer mentioned in the forward. He designs the total data system to operate within its control limits, and continuously monitors and varies data system elements to ascertain that control is maintained at an autonomic level. Data system elements include data sensors, the data paths and the data base actuator. A major system designer responsibility which has technical and political implications is the selection of, and total system accuracy distribution among, data sensors and actuators to meet a total system accuracy figure. Another function of the system engineer is to house all detail knowledge about the system and its capabilities in a single individual or function for access by anyone. The Data Administrator is a Data System Engineer.

Data Base Management

Data base management must first define the boundaries of data base content. Too much data leads to unnecessary data control problems and data structuring becomes unnecessarily difficult. Data access frequency statistics can indicate when it is feasible to move data to a slower storage media, and data compression techniques can reduce the size of back-up files and off-line storage. Often statistical or arithmetic techniques can be used to derive data from other data in the data base thus reducing total content.

The real objective of data management is control of data sources. A software product called the data dictionary presents to the Data Administrator a clear, unequivocal description of the data sources and the uses to which data is put. The objective of this study is to control data redundancies, but not data duplication as duplication is sometimes necessary to meet access response objectives.

Access response is usually met by attempting to match the data base structure by category to the firm's operational-level structure, then operational-level response time will be minimized. Response requirements at higher levels extend to days, weeks and months and therefore, are not critical. The reason the data base structure-firm data structure match is never met is due to the discrete nature of storage devices and the hierarchical storage and machine structures, the result of a huge difference between problem time and machine time.

Errors in a dynamic data base are inevitable, but they must be controlled. Data distributions may be estimated and extremes flagged for closer scrutiny. Regression may be used to estimate one variable's distribution from a group of others, and the difference between the actual and estimated distributions compared. Correlation techniques may show that various data-items are highly correlated, if the correlations change significantly over time, it may be due to accuracy degradation.

The security of data at present is not currently a problem as evidenced by the general lack of interest in the security market by manufacturers (with the exception of IBM) and users alike. However, interest will grow as data management system applications encompass the planning levels. If an extensive security system is implemented, control should be external to the main computer and sufficiently simple that one person can examine all possible system ramifications. Security is obtained by variety, not complexity.

The Data Administrator data base management function is simplified by data management software which perform many of his tedious tasks. An incredibly tiresome task which cannot be automated is the elimination of data redundancies, it is especially heartbreaking during the initial implementation of the DBMS.

Data Input System

As data base management system technology advances provide the environment for growth in number and variety of computer-based information system applications, the effect of incomplete, inaccurate, untimely data is heightened. Once unreliable data enters the data base, it may over time, trigger other data and data relationships which must later be corrected. Late data causes similar problems because timely linkages may not have been generated, the data to have been linked now being nonexistent or recently restructured. It is for these reasons that Data Administrator data accuracy and integrity must begin with the input system.

To increase data accuracy and reliability, data conversion and accuracy checks are being moved to the data source. Several studies by H. Sunderland reveal that 90% of all errors occur between the data source and data conversion to computer-readable form.[2] To better control data accuracy, distributed intelligence and redundant data input systems to back-up the primary input stream are being developed. These systems include remote data entry by intelligent and point-of-sale terminals, minicomputer-controlled key-to-disk systems and optical character recognition technology.

Currently, there is no theory for the design of the data input system, however, the configuration of the firm's data sources, the nature of the man-machine interface possibilities, and the variety of back-up systems determine which data conversion devices will be selected. General guidelines for device selection center on data volume, data variety and the ability of the conversion source operator. A large volume of standardized data, located at a single input source would require a key-to-disk system, an optical character recognition system, or an integration of the two systems, depending upon data source amenability to character standardization. Intelligent terminals are best suited for low volume, nonstandard input by a source document clerk.

A general characteristic of all the above systems is that data is monitored by the input device, and most errors are corrected on-line before they can penetrate the system. A secondary characteristic is the automatic collection of error statistics to aid source document design and better manage the input operation. A tertiary feature is a reduction in the amount of paper handled in the organization.

Data Communication

A general characteristic of the data input system is that it is moving closer to the input sensor, the data communication system often is the link connecting the input sensors to the data base. Geographic dispersion of the information system users is also becoming more common. Hence, in order to assign a data accuracy, integrity and security figure to each data-item, the Data Administrator must understand the data communication system and its weak points in sufficient detail to take proper measures to secure data from harm.

Data Administrator data communication knowledge must include carrier system capability, transmission techniques and error detection/correction techniques employed. Since error correction is usually <u>not</u> a part of the common carrier system, this must be implemented at the common carrier user level. A concatenation of error detection/correction codes within codes which approaches the Shannon limit for long codes is presented in Chapter IV. Most long haul common carrier systems are relatively error-free, it is at the local level that old mechanical equipment induce errors.

Currently, few communication systems implement anything more complex than parity checking. These routines as well as all other software routines and tables of terminal locations are stored in and processed by the mainframe. However, just as input system processing is moving out of the mainframe, so is the communications processing. The result is a more flexible system of independent subsystems communicating in a fail-soft manner. If the mainframe experiences difficulty, often data can continue to be collected with no loss in integrity. If the communication link goes down and a minicomputer-based operational data base exists at remote locations, critical processing necessary to maintain operational control can continue.

The planning and design of a data communication system should be analyzed along five dimensions:[3] geographic dispersion, whether it be plantwide, countrywide or worldwide; volume of data transmitted and its time pattern, since line quality is heavily dependent upon facility load; amount of manipulation required, processing and compacting data before transmission saves line costs; the decision point response requirement, geographically distributed data bases enhance critical operational system response and provide a fail-soft against line failure; and network availability to include balanced planning as protection against system overloads.

Hardware network planning is facilitated by commercial network optimizers which attempt to create a balanced, cost-effective system by indicating specific data routes and the location of multiplexors and concentrators. These optimizers, however, currently have limited capabilities. They do not allow intermixing of common carrier routes within a single data system and circuitous but cheaper paths through the common carrier network.

The above should _not_ imply that the Data Administrator should have data communication system _design_ responsibility. He must understand the function and tools of the data communications designer, but Data Administrator responsibility is to control data system design of which the data communication system is a part. The Data Administrator insures that the total data system is properly balanced, to include the input system, the output system, the structure of the corporate data base and the data communication linkage between these systems. The next section will present a means to control the design and operation of these systems to provide total data control.

Data System Control

Data system control is achieved by measuring the efficiency and the effectiveness of data systems. Efficiency refers to how well a system performs, and effectiveness refers to how well a system's design intention was met. Measurement may be static or dynamic. Static measures include using instruction mixes, kernel programs, benchmarks, synthetic programs and simulation to examine a pre-implementation system or a change to an existing system. Dynamic measures involve the use of hardware and software monitors to examine the "live" operation of an implemented system.

Static measures are useful in system design and equipment selection. However, these analyses are limited by the assumption that present and future system load and job mix are known with certainty and the exact multiprogramming and communications environment can be precisely duplicated.

Dynamic monitoring of the present system is often helpful in specifying load at any processing point. Once present and future loads are specified, the benchmark is the most popular analysis tool because it requires no new programming, only an intelligent selection of an active job mix and data.

Tuning of large and complex computer-based information systems is performed by monitoring ongoing activities. Data acquired from software and hardware monitors point out system bottlenecks. However, such data should not be collected randomly, but with some hypothesis or model in mind, this facilitates data reduction at the analysis stage. The minicomputer-based hybrid monitor, combining the attributes of a hardware and a software monitor provides minimum job stream interference while being able to relate collected data to operating software and is thus evolving as the analysis tool for tuning large complex systems.

The Data Administrator

In conclusion I see the Data Administrator as a Data System Engineer of corporate vice presidential rank. His data responsibility resides in and emanates from the data base and follows the flow of data through the organization from input sensor to data user regardless of geographic dispersion. External data source selection is also his responsibility. Interconnected data utilities and commercial hardware and software systems are being developed and will continue to evolve to make the Data Administrator's task more amenable to human function.

The Data Administrator assigns an accuracy and security measure to each data-item and then monitors and controls the data system to ascertain that data is not degraded below this limit. He selects sensors and assigns data channel accuracies to the data input system and the data communication system and then monitors the entire data system to determine that a balanced, cost-effective data system is achieved within the accuracy and security requirements specified for each data-item.

Since each data-item has associated with it an accuracy figure, management science modeling is more accurate, because a better match can be obtained between data accuracy and model sophistication. Modeling efficiency is enhanced, but the management prerogative of selecting the model to be used and the modeling objectives is unchanged.

This dissertation presented general technical characteristics and considerations of the Data Administrator. Specialized considerations and organization and human implications of the function are a subject of future research.

Footnotes

1. J.P. Fry, "Where We Are Going vs. Where We Should Be Going," and E.H. Sibley, "Some Practical Suggestions," Special Session on Managing the Impact of Generalized Data Bases, National Computer Conference, New York, 1973, to be published by the Special Interest Group in Business Data Processing of the Association for Computing Machinery, New York.

2. "Users Ride 'Intelligent' Bandwagon," Computerworld (April 25, 1973), p. 11.

3. H.B. Becker, "Information Network Design Can Be Simplified Step-By-Step," Computer Decisions (October, 1972), p. 14.

APPENDIX A

APPENDIX A

MANAGEMENT INFORMATION SYSTEMS--A PRESPECTIVE

1. MIS--A GENERATIONAL PRESPECTIVE

1.1 First Generation

Robert I. Benjamin, writing in the Twenty-fifth anniversary issue of the ACM Communications, puts forth a generational prespective of information system development which parallels the hardware and software generations [1]. The first generation occurred between 1940 and 1950 and allowed little information system development.

The major use of machines during this period was to calculate mathematical tables. The problem of developing accurate mathematical tables had existed for centuries and attracted the intellects of Prony (1784) and Babbage (1800). Prony attacked the problem as one of organization. The task was divided along a hierarchy of computation, the highest level consisting of outstanding mathematicians and the lowest level consisting of a multitude of people who knew only how to add or subtract. Babbage, influenced by the Jacquard Loom (1801), attached the problem mechanically, designing a calculating machine which contained the algorithm, thus eliminating the human calculators. However, due to mechanical problems of backlash in gears and human problems instigated by Babbage's irascible nature, the project was never completed. It was not until the MARK I was designed in 1943 by Howard Aiken of Harvard that Babbage's papers were discovered. The MARK I looked remarkably similar to Babbage's machine with the exception that telephone relays were used in its construction, circumventing the mechanical problems faced by Babbage.

The ENIAC, constructed at the University of Pennsylvania by Eckert and Mauchley, was the first "electronic" calculating machine. It operated at electronic speeds by replacing moving relays with eighteen thousand vacuum tubes. Relays were slow and subject to wear and sticking. Vacuum tubes were much faster but also had a predictable failure rate. The major problem was in keeping the machine operating. It was difficult to tell when a tube failure had occurred, and many times it was not until days later that a failure became evident. Locating the source of each failure was a major problem sometimes taking days. In all, there were periods where not more than five percent of available machine time resulted in useful work [2].

The great problems during this generation consisted of internal machine design and reliability. Von Neuman made a significant contribution in these areas by suggesting that machine speed could be increased if (1) the program and data were stored in the computer memory, thus reducing electronic circuit path length, and (2) machine reliability could be improved considerably if the binary number system were used, since electronic circuits operate most reliably when only two states need be recognized--on or off.

1.2 Second Generation

The advent of transistor technology ushered in the second generation information system, occurring between 1950 and 1964. Since transistors do not wear out, but fail randomly, the probability of failure being as great when the machine is new as when it is old, computer reliability was greatly enhanced allowing their use in business applications.

Often applications which used punched card equipment were exactly duplicated on the computer. Separate systems existed for payroll, inventory, customer billing and stockholder records. As development of many such systems progressed, a few system designers recognized that some data was used as input to more than one system--data was beginning to be recognized as a basic organizational resource. Increasingly, data borrowing among functional systems became prevalent.

Major machine problems faced during this time centered on programmer dominance of a hardware-memory limited machine wrested through the sheer blood and sweat of efficient machine level coding. However, once this dominance was attained over one machine configuration, major changes were required to modify the software system or obtain improved hardware. Even though a single hardware-software application set could be optimized, hardware-software dependence hung as a millstone about the neck of every systems programmer.

1.3 Third Generation

The hardware-software dependency syndrome was eased to some extent by third generation systems during the 1964 to 1968 timeframe. Machine mastery was now assisted by the widespread use of procedural languages such as COBOL and a reliance on operating systems to control the hardware. Unburdened by stringent machine problems, the systems designers made halting attempts at system integration to overcome total system suboptimization. Though the classic article on system suboptimization was written in 1953 by Hitch [3], it was not until the mid-sixties that analysts thought they possessed the technology to attempt to dominate suboptimization by controlling conflicts between functional systems--such as the classic conflict between production control and inventory control. At the same time there was pressure to standardize systems in large, growing corporations, which pressed for the development of a centralized systems development function [4].

Thus the easing of hardware and software constraints resulted in the development of integrated systems of larger scope. However, these large multifunctional systems were costly to develop and required a long time to install. The result of these experiences was the description of the development life cycle, and a number of software system design aids, such as mechanized flowchart preparation and documentation.

1.4 Present Generation

Once again the solution of one class of problems, hardware-software dependency, laid bare another problem, that of system-data dependency. Systems, heretofore, were designed by first designing the specific output report, then working backward to identify the required inputs. This approach is acceptable if the system is well defined and unchanging over time, hardly the description of one of today's large organizations.

Clearly, a solution must evolve from a new method of viewing the organization. The concept of a multilevel management system provides the structure for such a solution. This concept recognizes a technical or operational core of the organization which can be controlled by deterministic methods and the decision paths clearly specified with certainty, i.e. programmed. Surrounding the core and buffering it from externalities is the tactical management system which allows the operational system to function in a deterministic environment. The strategic system surrounds the foregoing to specify organizational goals and control policy. These systems form an interlocking, overlapping hierarchy whose connections sometimes look more like a network than a hierarchy. However, networks are difficult to understand and analyze so a hierarchy is assumed, and is in fact a close enough approximation or model of reality.

The organization can then be described as a joining together of two systems--the management system and the information system. The objective of the management system is to present an orderly and systematic method for controlling and directing information, i.e. it is the decision structure of the organization [5]. The information system has the basic objective of making available a broad base of comprehensive information, such that timely information flows to each decision point in sufficient quantity and quality to allow the best decision to be made, i.e. the selection of the alternative having the highest probability of success.

Integrated with this model is the psychology of each manager. Each manager because of personality differences and management style develops his own unique model of the organization and the information he requires [6]. Second and third generational systems required extensive reprogramming to include the new input data necessary to generate the new output report. This is a frustrating state of events resulting in constant system change of major proportions.

There is a better way. Instead of beginning by defining specific output reports (which are subject to change) one may start by defining all the data the corporation needs to perform its function at all hierarchical levels. Such data may exist at the elemental level within the corporation's computer memory or exist in some external information utility. The point is that data is independent of any application and is treated as a corporate resource on a level with men, materials and capital. This is the data base concept. If procedures and data are independent, the models can change while the data continues to reflect occurrences internal to and external to the organization. Furthermore, the data is such that it can be utilized by any manager's model.

Many new problems arise with the data base concept, such as what data should be included in the data base, and who is responsible for data input, data security and data integrity. To solve these problems new organizational forms will arise, new functions such as Data Administration will be created, and new tools such as the data dictionary, and the data description and data manipulation languages will be invented.

Even given the problems this approach presents, in the long run it is a better procedure. For when change in output reports is requested or new applications are necessitated, it is a relatively easy task to supply new information, because the data already exists in the data base and need not be captured.

2. INFORMATION SYSTEM DESIGN

2.1 Two Approaches

The data base concept is the culmination of information system development. It currently is being implemented to various degrees in every major corporation, however, controversy still exists over the design process of the information system.

Two approaches or philosophies of information system design have been put forth, the top-down approach and the bottom-up approach. The top-down approach requires that the overall system objectives be determined and the system be defined by moving down one level at a time until the level of basic data is achieved. The bottom-up approach assumes that high level goals and objectives cannot be determined in sufficient detail so that one must start at the level of basic information and move through the organizational systems in a closed loop manner until all systems have been defined. However, there is evidence that one evaluates proposed projects along two dimensions: (1) the extent that the project integrates several functional applications into one application and (2) the degree to which each consolidates data to serve different levels of management. Thus it appears that the MIS will evolve along both horizontal (bottom-up) and vertical (top-down) dimensions as a continuing rationalization of earlier generations [7].

2.2 Mastery of the Organization--A Man-Machine Symbiosis

It would appear that the central problem for MIS immediately is to master the organization. With the acceptance of the data base concept, information has been recognized as a basic resource which allows the vertical and lateral integration of operations. The consistency and standardization of this information is the cornerstone of this effort. Finally, the information system must be recognized as a man-machine symbiosis, with each performing those tasks they do best. The machine, for example, is capable of performing calculations tirelessly with great accuracy while maintaining an excellent short term memory. Man on the other hand has a fantastic ability to see patterns in complexity while maintaining an excellent long term memory. The organization should then be viewed as a closed man-machine system, much like a pilot-aircraft system.

Three technologies define the mechanistic part of the system, computer technology, which allows the processing of large volumes of data, management science, which allows the reduction of data to control variety, and systems analysis, which integrates men, machines and organizational functions.

The systems analyst must see himself as an ecologist, whose responsibility it is to sustain and improve the organization. One important aspect is to predict how a change in one subsystem may reverberate throughout the organization. As Edmond Dewan points out, in large systems certain types of instability may elude detection since they may be a property of the system as a whole--and not of any individual component or subsystem [8]. Therefore, it may not be possible to anticipate all changes. Keenan Sahin proposes the addition of a noise factor to the information system as an evolutionary growth mechanism to match system environmental changes or drift which might be due to a change in either the total system environment or a change in a subsystem [9]. There is theory to support this assumption both in biology and psychology (Bem's Belief Structure). Benjamin suggests that different rates of change would be a function of the hierarchical level, the operational level varying slowly and the strategic level varying most rapidly. The possibility of unpredicted behavior, the addition of noise as a controlling element and the realization of differing information rates of feedback time lags at different hierarchical levels within the organization is integrated with the concept of matching variety with variety. Any management information system must be designed to exactly match the environment in both concept and form. Only variety can destroy variety. The control system must be sufficiently robust to match the environmental possibilities.

3. DESIGN OF THE MANAGEMENT INFORMATION SYSTEM

3.1 Design Considerations

With this overview of the information system, let us proceed to define how one might design an information system. The underlying concept to this design will be the development of systems which serve more of the real needs of management, and do so more efficiently than a series of individual systems. They must be easy to modify and change and must meet their implementation schedule. Furthermore, such systems must present results to people so as to maximize efficiency of the man-machine combination. The overall viewpoint is that of a system with key factors being the conservation and management of information implemented through the economical collection and efficient storage of accurate, timely and meaningful data, which mirrors the condition of the environment. A second design consideration is the data base concept in which each element of data is collected once and stored in one place. A third design consideration is programmed change which must be flexible, to meet a wide variety of circumstances, and adaptable, to meet major environmental changes with a minimum of modification. The criteria for adaptability and flexibility will be met if the information system follows the basic structure of the organization and data is collected and structured at its most fundamental level.

3.2 The Systems Concept

These design considerations must be integrated via the systems concept. The general implication of a closed system in a cybernetic sense is a series of interconnected elements, where changes in external environment cause elements to react cooperatively to maintain system objectives [10]. In a hierarchy of systems, one system may be the environment of another. A major problem is that the character of two subsystems may be transformed when they are combined into one system.

3.2.1 System Hierarchy

Mesdrovic and Macko put forth five propositions in describing a hierarchical system [11]. First, the level at which the system is described depends upon the observer, his knowledge and interest in the system. A plant manager might see the company as a collection of plants, while a marketing executive might define the company in terms of its product lines. The basic building block determines in what context the system will be analyzed. Second, the contexts are not in general mutually related, and the principles or laws used to characterize the system on any level cannot in general be derived from principles used on other levels. Third, there exists an asymmetrical interdependence between functioning of the system on different levels. Thus, the proper functioning of the system on any level appears as constraints on the operations on lower levels [12]. Fourth, each stratum has its own set of terms, concepts and principles and what is considered a system is different on each level. Furthermore, there is a hierarchy of languages in which they are described. As a rule, the description on any level is less detailed than on the lower level. A subsystem on a given level is a system on a lower level. Fifth, starting from any given stratum, understanding of a system increases by crossing stratum; moving down a hierarchy one obtains a deeper understanding of its significance. By reference to lower levels one is able to explain more precisely and in more detail how the system functions to carry on a certain operation rather than which principles determine the particular operation to be performed.

3.2.2 The Problem of System Overlap

The above conditions for a hierarchical structure could be clearly represented by a tree-like structure wherein each element of the system belongs to only one subsystem at each hierarchical level. It is also possible, however, that at a given level system, elements may belong to more than one subsystem and are, therefore, connected to the highest level by more than one hierarchical path.

Schlicta suggests three stringent requirements which taken in themselves will avoid all hierarchical overlap [13]: (1) Subordination, in which the forces binding the elements to the first level subsystem must be weaker than the forces joining the subsystems into large systems, (2) Localization, in which the forces attracting individual elements to each other must be so specialized that there is no appreciable interaction between elements of different systems, and (3) Monodimensionality, in which every element and subsystem can only combine in one way to form larger systems. However, if the system has a plurality of functions, interests or aspects, a multidimensional type of overlap is inevitable.

The preceding problem may be eased by the manner in which a complex structure is dissected into its subparts.

The trick is to follow the "natural interfaces," severing a minimum of connections in isolating the subcomponents, otherwise, the decomposition may be confusing. These "natural interfaces" are identifiable either by the occurrence of a steep decrement in the number or strength of linkages crossing them, as developed by Simon in the concept of near decomposibility [14], or through the existence of some form of closure. Closure is cyclical in the sense that some parameter follows a path that periodically returns to previously assumed values--life itself or addition modolo 10 are examples. Thus a corporation's subsystems may be recognized by identification of various feedback loops.

The problem is to identify these feedback loops--a skill which requires a certain amount of artistic talent. The artist has a great sense of the relationship of parts to wholes. To the artist there are basically few unit patterns involved in any combination. Practically everything can be divided into isolated granular areas with concomitant connectivity of the boundaries, and the branching connectivity of linear growing tree-like forms. These same forms may be repeated in rocks in the foreground, boulders in the midground and mountains in the distance. The simple local relationships repeat at higher levels to diversity without limit.

3.2.3 Variety and Constraint

The prime consideration in specifying a hierarchy is the reduction of the amount of variety presented by the system to enable the system behavior to be predicted and controlled [15]. The psychiatrist, W. Ross Ashby, has presented the Law of Requisite Variety, which states that in order to control or predict the behavior of a system the controller must possess at least as much variety as the system [16]. This state of affairs may be reached by either increasing the variety of the controller or reducing the variety of the system to be controlled. All quantitative techniques serve to reduce the variety the controller sees so that system behavior may be predicted and controlled. Every model, every law of nature, humanistic or physical is a constraint on the variety which may be assumed to exist in nature.

3.3 Information Mainstream

3.3.1 Definition

A general variety reducing model which can be applied to any organization is the information mainstream model. The information mainstream touches every subsystem and is the key process in the organization, which may be material movement through a plant or paper processing through a service organization. Mainstream information originates at many sources and must be transformed and transmitted to many decision points. Flows are vertical and horizontal within the hierarchy. Vertical flows are summarized at higher levels and horizontal flows provide specialized departmental needs as a natural by-product of the basic information flow.

3.3.2 Data Structure

The basic information flow of an organization is defined by some transformation function which accepts primitive data as input and generates derived information as output. This function is called the organization's data structure and consists of all the subsystem transformation processes defined by the system analyst. If the subsystems are properly defined, material flow and constraints will be independent making it necessary to only define the data variables appearing at the subsystem boundaries. The data variables existent at these boundaries are stored in the corporate data base or memory in elemental form so they can be used as input data to any model perceived by management. The hardware storage structure of the data is called the physical data structure and is perceived to be a one-to-one and onto mapping of the organization's data structure onto the hardware. However, due to physical constraints this ideal is not achieved in practice and much effort is expended in continuously optimizing storage structure as information system speed and accuracy requirements change to meet a fluctuating organizational environment.

3.3.3 Information Sensing

The environment of each organiztional subsystem is sensed at the subsystem boundaries. At each boundary several elemental activities or events occur which are physical in nature and whose occurrence is easily recognized. These activities must be selected such that they are sensed at their most basic level, at natural breaks in the process, as close to their occurrence in time and place as feasible. If activities are sensed at their most basic level, the elemental data can serve as input to any level of subsystem. If sensing occurs at natural breaks in the process, errors of omission will be minimized. Furthermore, if the activity is reported as close to their occurrence in time and place as feasible, error detection and correction will be further enhanced. The longer an error goes undetected, the farther it penetrates the system, and the more difficult it is to correct, or even determine what the error was. It is far more reasonable to assume that the one who input the error knows what the correct response should be. Therefore, a general statement of data responsibility may be put forth: every level of management must be directly responsible for that portion of the system sensor under their jurisdiction. Data errors and inaccuracies are the nemesis of the decision maker, poor data can only produce poor decisions, irrespective of the sophistication of the model used.

In addition to the requirement for accurate data, there is a requirement for a rigorous definition of the data elements so they may be utilized by the entire system. This can best be controlled by a centralized data dictionary and is the first step any organization must take in implementing the data base concept. The function of controlling the data base in this facility comes under the purview of a Data Administrator whose responsibilities and functions are still evolving [17].

3.3.4 Information Analysis

Once the data base has been defined and the security and integrity of its contents established, its contents may be used in modeling and information analysis. Several difficulties arise because of data time dimension, variable value and the problem of data identification. Data is used and then discarded as a physical entity it describes leaves the system--the job priority of a completed task for example. The same data may be used at different points in time with different values associated with its use--production costs may be used in a decision model, summarized or audited. Finally, as information flows through the organization it is converted and loses its identity, making analysis difficult.

Irrespective of these difficulties, the most important implication of the data base concept is for on-line planning and exception reporting. Sara Read of Pillsbury has stated that exception reporting has been only a dream until the advent of the corporate data base and on-line inquiry [18]. Heretofore, one could not follow the chain of events defined by the data structure to the problem source. The result was voluminous output so that relationships could be traced manually.

The crux of computerized information analysis is to assign to the computer the repetitive tasks that follow definable rules and to humans the tasks involving creativity, imagination or complex logic. People are adept at recognizing patterns in graphic data which would otherwise elude all but the most careful statistical analysis.

4. IMPLEMENTATION OF THE M.I.S.

The implementation of the management information system must lie initially in the mainline operations of the organization [19]. This is where the bulk of the expenses occur and from which the main income results. The implementation procedure must involve a series of smaller projects embedded in a larger long-range plan. One would then observe an almost continual progress with one project after another. The undertaking of a large project as a series of smaller projects is not new. What is new is that each project is aimed at helping the operating people, but with smaller projects and smaller risks.

In this design approach there is a clear goal in mind and the sequence of projects or events is known. But follow on projects are not known in detail, they will be influenced by the results and experiences of earlier projects. Cost estimates are easier to predict with small projects and the accuracy of such estimates improve over time. The third and a half hardware-software generation makes this approach feasible, since equipment can be expanded in small increments. Other benefits of this approach is that management can see results sooner and is more willing to become involved in projects of short duration. Also, since fewer people are involved, each individual can see his part of the total effort resulting in more personal satisfaction and less pressure. And because one person can view the system in its entirety, logical flaws can be detected. Project management is easier because the schedule is shorter and involves fewer tasks. Cost and schedule slippages are detected earlier, corrected easier and have less severe impact.

The environment the user sees should also be changed in small steps. For example, it may be possible to add or change on-line terminals and the system while keeping the same report format for awhile, thus presenting to the user a constant rate of change rather than a large fluctuation which may cause unnecessary consternation. It must be remembered that a system is only successful if the users <u>want</u> the system to be successful because they think they need it. It is therefore necessary that a good implementation track record be obtained immediately by initially selecting an area with the highest chance of success--a well defined process where users are willing to cooperate with enthusiasm.

The result of this implementation procedure is smooth, steady, evolutionary growth of the management information system as first mainline elements are integrated with branching applications following. This implementation concept is aided by the implementation of the data base concept and a data management system which allows the data base to grow with the mainline effort and the applications to grow independently at their own rates.

FOOTNOTES

1. Robert I. Benjamin, "A Generational Prespective of Information System Development," Communications of the Association for Computing Machinery, Volume 15, Number 7 (July, 1972), pp. 640-643.

2. Frantz L. Alt, "Archaeology of Computers--Reminiscences, 1945-1947," Communications of the Association for Computing Machinery, Volume 15, Number 7 (July, 1972), pp. 693-694.

3. Charles Hitch, "Suboptimization in Operations Problems," Journal of the Operations Research Society of America, Volume 1, Number 3 (May, 1953), pp. 87-102.

4. F. G. Withington, "Data Processing: Evolving Place in the Organization," Datamation (July, 1965), pp. 58-68.

5. Herbert Simon, Administrative Behavior, (New York: The Free Press, 1967).

6. Richard O. Mason and Ian I. Mitroff, "A Program for Research on Management Information Systems," Presented at the ORSA-TIMS-AIIE Joint National Meeting, Atlantic City, New Jersey, Novémber 8-10, 1972.

7. Robert I. Benjamin, "A Generational Prespective of Information System Development," Communications of the Association for Computing Machinery, Volume 15, Number 7 (July, 1972), p. 642.

8. Edmond M. Dewan, editor, Cybernetics and the Management of Large Systems, (New York: Spartan Books, 1969).

9. Keenan Sahin, "Towards Associative Information Systems," Presented at the Association for Computing Machinery Annual Conference, Boston, Mass., August 16, 1972.

10. A. D. Hall, A Methodology for Systems Engineering (Princeton: D. Van Nostrand Company, Inc., 1962), p. 61.

11. Mesdrovic and D. Macko, "Foundations for Scientific Theory of Hierarchical Systems," Hierarchical Structures, edited by L.L. Whyte, A.G. Wilson, and Donna Wilson (New York: American Elsevier Publishing Company, 1969), pp. 29-51.

12. Polanyi, "Life's Irreducible Structure," Science, Volume 160, pp. 1308-1312.

13. Paul J. Schlichta, "Overlap in Hierarchical Structures," Hierarchical Structures, edited by L.L. Whyte, A.G. Wilson, and Donna Wilson (New York: American Elsevier Publishing Company, 1969), p. 142.

14. Herbert Simon, "The Architecture of Complexity," Proceedings of the American Philosophical Society, Volume 106, pp. 467-482.

15. James C. Emery, Organizational Planning and Control Systems (New York: The Macmillan Company, 1969), p. 20.

16. W. Ross Ashby, An Introduction to Cybernetics, Science Editions (New York: John Wiley and Sons, Inc., 1963), p. 206.

17. R. G. Canning, "The 'Data Administrator' Function," EDP Analyzer, Volume 10, Number 11 (November, 1972).

18. Sara Read, "Issues in Data Base Technology," Presented at the Association for Computing Machinery Annual Conference, Boston, Mass., August 16, 1972.

19. R. G. Canning, "Progressive Fast Response Systems," EDP Analyzer, Volume 8, Number 10 (October, 1970), pp. 9-13.

APPENDIX B

APPENDIX B

DATA ADMINISTRATOR DESCRIPTIONS

<u>Topical Outline of the CODASYL Systems Committee Draft on the "Data Administrator Function."</u> [1]

Monitoring Requirements.

- Detection of loss or the risk of loss.
  - Monitoring to detect hardware failure.
  - Monitoring of integrity.
  - Monitoring to detect attempts to violate data security.
- Positive pursuit of quality and efficiency.
  - Quality Control monitoring.
  - Monitoring the efficiency of storage structure.

Support facilities to take action.

- Recovery from loss.
  - Restart and recovery backup.
- Actions available to improve integrity.
  - Integrity preservation administration.
- Action to preserve security.
  - Security preservation administration.
- Quality Improvement facilities.
  - Activities to improve quality and efficiency.
  - Quality improvement specifics.
- Capabilities to reallocate data to storage structure.
  - Storage restructuring for efficiency.

Capabilities to reorganize data structure.

Rationalizing of data structure.

Facilities to notify users.

Monitoring data provided to users.

Vesting of data administration function.

Detection and response to hardware failures and data loss.

Detection and response to failure of data integrity.

Detection and response to attempted breach of security.

Monitoring and action in quality improvement.

Monitoring and action in storage structure efficiency.

Monitoring and action in data structure efficiency.

Education of human agents.

Topical Outline of the CMSAG, "Data Administration" Working Paper. [2]

Background.

Purpose.

Scope.

Objectives.

Establish the data dictionary.
Maintain the data dictionary.
Publish the data dictionary.
Periodically review physical Data Structures.
Review Logical Data Structures.
Determine Security and Integrity Requirements.
Unify Code Structures.
Provide Technical Consulting to Users, Programmers and Operations.

Justification.

Recommendations.

Future Reports on Data Administration.

Comments of James P. Fry [3]

James P. Fry sees the Data Administrator's role as requiring qualitative and quantitative tools. Qualitative tools are the ability to get along with users, and understand the nature of the data being processed. The Quantitative tools require:

(1) Knowledge of the system and data base management techniques.

(2) Analytic techniques for designing files and data bases.

The knowledge required about the system includes such things as performance characteristics of access methods, and the internal organizations of data base management systems (DBMS). As far as the DBMS is concerned the Data Administrator must know the effective way of defining the files logically and physically to the system. To achieve this he must have a knowledge of the built-in implementation tradeoffs of the DBMS.

Comments of John K. Lyon [4]

John K. Lyon sees a parallel between data processing evolution and the evolution of society. Early data processors could be rugged individualists like the early pioneers who prided themselves in their self-sufficiency and independence. Data processing evolution from independent systems to the integrated data base is akin to societal shift from individuals to communities. In the new data community it is up to the Data Administrator to police and enforce rules pertaining to data and to provide community services.

The Data Administrator must be a corporate watchdog, monitoring information system performance. Besides policing the data he must act as an army chaplin, advising those in need of guidance and offering assurance to the inexperienced and the timid.

The extent of the Data Administrator's duties, his staff size and its degree of specialization will depend on (1) company size, (2) company management practice and (3) the current state of integration of company information systems.

Comments of R. F. Schubert [5]

The Data Administrator is one or more technical experts who are knowledgeable in data base design and creation, operation of the data base management system, and the use of one or more data manipulation languages. Data Administration duties are to:

(1) Work with systems analysts to determine application data requirements.
(2) Aid programmers in the most effective techniques in the use of the data manipulation language (DML).
(3) Create subschemas as required by applications.
(4) Maintain documentation of the data base schema and document subschemas for programmer and analyst use.
(5) Establish appropriate operating recovery and roll-back procedures to preserve the integrity of the data base in the event of either hardware or soft-

ware failure.

(6) Evaluate data base loading and program performance characteristics to recommend improvements.

(7) Supervise the addition of new areas, data items, record types, and set types to the data base.

(8) Initiate data base restructuring whenever it is needed to provide additional physical space or changes in data structure.

(9) Establish appropriate constraints in the use of data manipulation (DML) statements for each subschema.

Comments of the CODASYL Data Base Task Group [6]

The data base is a compromise between the various needs of user applications. The function of Data Administration is the human activity of mediating these needs. Data Administrator activity may be described as organizing, monitoring and reorganizing. These functions must be performed in order to have a viable data base system.

Organizing

Build the schema and possibly the subschema.

(1) Employ a data structure that models the business.
Structural Levels: Area, Record, Set, Data-item, Data-aggregate.
Tool: Data Description language. (DDL).

(2) Assign Unique Names.
Tool: Data Dictionary.

(3) Select Search Strategies based on response requirements of users.

(4) Assign Privacy Locks and issue Privacy Keys to users based on the need to utilize particular data.

(5) Assign Areas to devices/media based on time/space requirements.

(6) Load the data base.

Monitoring

Monitor the data base for usage, response, privacy breach, and potential reorganization.

Tools: Logging facilities, Statistical sampling.

Reorganizing

Result of Data Base Growth and/or monitoring.

(1) Reassign areas to different devices/media.
(2) Change the schema and/or attributes of schema elements.
(3) Change the data base to reflect changes in schema.
(4) Remove 'dead' records and compact space.

Comments of R. G. Canning [7]

Canning sees the Data Administrator function as being divided into six major types of activities: design, administration, operations, monitoring, auditing and system improvement.

Design Activities (includes maintenance of designed elements)

1. The Data Administrator should design the standard data definitions, the data dictionary and the data base itself, to meet conflicting user needs; design includes the data structure of the whole data base, the data structure as seen by application programs, the storage structure, and mappings between them; design also includes search strategies and access methods to be used--plus file membership rules, record relationships, defaults, and data compression techniques.

2. Define the rules of use of the data base and any access constraints; such constraints would apply to exclusive and shared use; rules would attempt to avoid concurrent updates and deadly embrace (where there is not enough of a resource available to be allocated to satisfy any job, and no job is able to run); access constraints should apply to source and object programs stored in the data base.

3. Design the security system to guard against: penetration, unauthorized update or copying, inadvertant disclosure, removal, or destruction; system may include locks and keys, logging, cryptography, etc.

4. Design the integrity system, to guard against inaccurate, invalid, or missing data, and to flag suspected data; system may include validation checks, logging, dumps, recovery system, etc.

5. Design the support software, for creating, maintaining, and reorganizing both the data base and the data dictionary.

<u>Administrative Activities</u>

1. The Data Administrator is the custodian of the corporation's mechanized data, both in the data base and in the older application-oriented files. Management sets the policies on disposition (copying, removal, destruction, etc.) and the Data Administrator interprets and administers those policies. The Data Administrator ascertains that all requirements on the retention of data (such as by government regulations) are met.

2. The administration of data standards, including the review and approval of new data definitions and the enforcement of data standards.

3. The administration of the security system, including: the assignment and subsequent modification of data locks and keys; machine room procedures; decisions on actions to take when security is breached; setting up security system with distributed responsibility and with checks and balances.

4. The maintenance of data base documentation, including directories, glossaries, cross-reference listings, and so on.

5. Communication with users of the data base, to provide assistance and guidance on use, to detect and correct user problems, to notify users of changes in system status, etc.

6. To conduct or to participate in the selection of hardware/software for data storage and retrieval.

Operations Activities

1. The Data Administrator is responsible for creating and reorganizing both the data dictionary and the data base. In creating the data base from multiple application-oriented files, he is responsible for getting all data conflicts cleared up (such as field values that should be the same but in fact are different). When the data directory is changed, the base must be revised accordingly.

2. The Data Administrator should initiate and/or control the data base integrity procedures, including logging, dumps, audit trails, checkpointing, and recovery procedures.

3. Control the use of the data base and restrict rights of use, as needed--such as during recovery, during periods of degraded performance, during testing of programs, and upon evidence of user conflicts, security breaches, etc.

4. Reallocate the files to alternate media, as required.

5. Exert some control over computer scheduling, so as to "schedule around a problem" and to provide for priority use of the data base.

Monitoring, Auditing, and System Improvement

1. Monitoring has to do with measuring various aspects of performance. The Data Administrator should define the rules for and initiate various types of monitoring. In addition to regular configuration performance monitoring, this activity should cover the monitoring of integrity routines invoked, security procedures invoked, response times achieved, error counts by type, and the efficiency of routines that make substantial use of resources.

2. Auditing has to do with determining compliance with established standards. The Data Administrator should audit quality and redundancy, documentation procedures, integrity assurance procedures, and security system procedures. It is likely that sampling techniques will be used for such audits.

3. The need for system improvements will become evident from a number of sources, such as the results of monitoring, results of audits, operational difficulties, and so on. The Data Administrator should be responsi-

ble for reviewing such results and difficulties and for initiating any needed data base improvement activities.

The data administration function is a complex one. It is unrealistic to expect that the function of data administrator can emerge full-grown overnight. Each company will have to select those activities that are most important to it and begin the data administration function with those activities. Gradually, the job can be enlarged as the function evolves within the company.

FOOTNOTES

1. "Data Administrator Functions," (July 8, 1970), CODASYL Systems Committee, W.H. Stieger, Chairperson, Chase Brass & Copper Company, 20600 Chagrin Blvd., Cleveland, Ohio 44122.

2. "Data Administration Working Paper," (April 21, 1972), Construction Management System Action Group, J.N.R. Bryan, Chairperson, Florida Power and Light Company, Box 3100, Miami, Florida 33169.

3. Personal correspondence with James P. Fry, Department of Industrial and Operations Engineering, College of Engineering, The University of Michigan, Ann Arbor, Michigan 48104.

4. J.K. Lyon, "The Role of the Data Base Administrator," Data Base, Vol. 3, No. 4 (Winter, 1971), p. 11-12.

5. R.F. Schubert, "Basic Concepts in Data Base Management Systems," Datamation (July, 1972), p. 47.

6. CODASYL Data Base Task Group April 1971 Report, (1133 Avenue of the Americas, New York: Association for Computing Machinery), p. 22-23.

7. R.G. Canning, "The 'Data Administrator' Function," Vol. 10, No. 11 (November, 1972), pp. 12-13.

APPENDIX C

APPENDIX C

VIRTUAL STORAGE

INTRODUCTION

As hardware has become more sophisticated and machine architecture more complex it has become increasingly difficult for a single programmer to code a procedure which will run efficiently in a given machine and load environment. In many large systems several programs may be running concurrently with an on-line system yielding a load whose affect on hardware resources is unknowable to the programmer. Hence, it is appropriate to define a system which will relieve the programmer of the burden of total system efficiency.

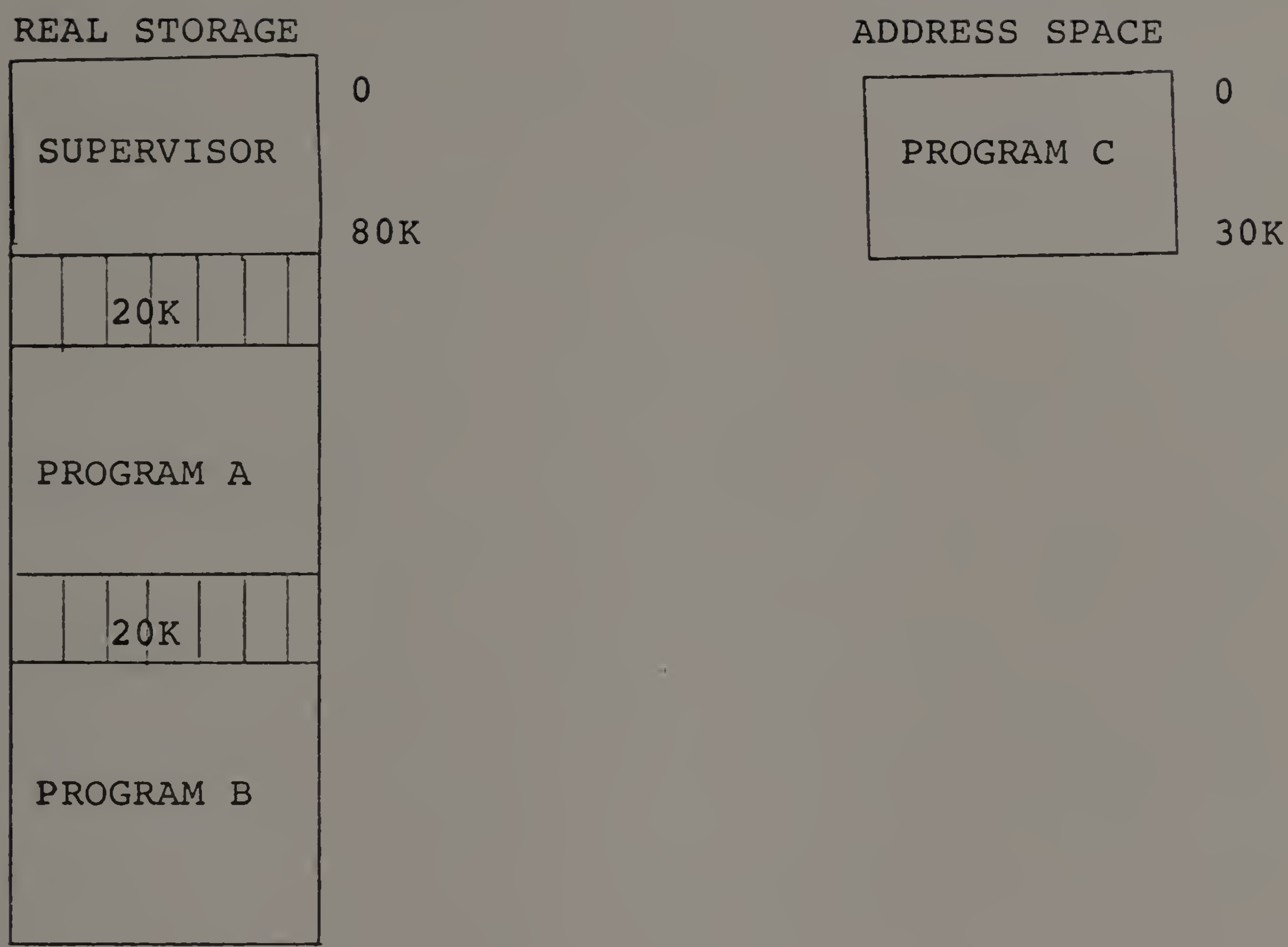
Fifteen years ago computers executed programs one-at-a-time, in sequence. Each program was set up and run independently by the system's operator. To reduce card handling and increase throughput an operating system was designed so that jobs could be grouped or batched. This eliminated the idle machine time required for operator intervention. If jobs were properly sequenced the operator could set up tapes and output forms for the next program while another program was executing. These operating systems also unburdened the programmer by providing I/O routines to handle read or write errors and data security. As CPU speeds increased and main storage became more abundant, operating systems were expanded to provide multiprogramming, the running of multiple jobs concurrently. By sharing the CPU, real storage and other system resources among active jobs, resource usage and total system efficiency measured by throughput was enhanced.

Virtual storage is an extension of the multiprogramming operating system which (1) provides to the programmer a much larger address space than actually exists in hardware and (2) dynamically allocates real storage to jobs, as required. To understand the effect of this concept the IBM System 370 implementation of virtual storage as adapted from Introduction to Virtual Storage in System/370 will be described. [1] The concepts of relocation, segmentation and paging and the operation of the virtual storage system will be depicted followed by an analysis of the impact on programming style and present systems.

RELOCATION

Relocation provides flexibility to compiler generated code in a multiprogramming environment. Compilers generate code which begins at real storage location zero. Several programs utilizing the same real storage locations cannot exist in core simultaneously. Hence, the entire program is relocated in an empty contiguous area of real memory by either software translation of the addresses before execution (static relocation) or hardware translation of the address during execution (dynamic relocation). If a program is translated before execution it is bound to its real storage locations until execution ends. This is the case in OS/360 multiprogramming. The result may be the fragmentation of real storage in which fragments of memory between programs are wasted and other programs are made to wait because a sufficient amount of contiguous storage is not available. For example, in Figure 1, program C requires 30K but must wait, though 40K is available, because program A and program B are not ideally located in core to permit program C to be inserted.

In order for a program to be executed it is necessary that the program address structure be contiguous, i.e. continuous from location zero through program end. With dynamic relocation, since address translation occurs at run time it is possible to give the appearance of continguous locations though the program may actually exist in discontinuous segments of real memory. The program is divided into segments by the operating system and compiled. Each segment has a beginning address of zero. A pointer table (called a Segment Table) which relates segment name to beginning real location is set up, this is called "mapping" because the segment table maps the program onto real storage. A Segment Table Origin Register (STOR) is also required to point to the beginning of the segment table in real storage. For example, in Figure 2 the segment table origin register con-



STATIC RELOCATION REQUIRES PROGRAM C TO WAIT FOR 30K OF CONTIGUOUS CORE

FIGURE 1

tains the address 68,000 which points to the segment table origin. The segment table then points to the beginning of each segment. Since all addresses are relative they must be translated to physical addresses by the Dynamic Address Translation (DAT) hardware feature. This feature accepts the segment address and relative address within the segment and then concatenates or links these addresses to yield the required physical address. Thus the only machine time required in address translation is the table access time. However, additional memory is required to maintain the tables in real memory.

SEGMENTATION AND PAGING

Segment and Page Tables

Segmentation is a better way to manage real storage. With segmentation less real storage fragmentation is achieved than in static relocation systems like OS on System/360. Segments are smaller than programs and, therefore, with segmentation, fragments are smaller, however, fragmentation does still exist. It is intuitively obvious that if the segments were made small enough, fragmentation would be minimized, but at the cost of tables of increased size with their inherent usage of real memory and system overhead. Therefore, to obtain an efficiency of real memory usage without increasing system overhead excessively a hierarchy of segmentation, called segmentation and paging is designed.

Each program is segmented by the operating system which then cuts each segment into smaller pieces called pages. (IBM Operating Systems utilize 2k or 4k pages.) Smaller pages produce better real storage utilization while larger pages enhance total I/O operations. [2]

The table reference structure is enhanced by the addition of a page table for each segment. The segment table then points to the beginning of the page table for that segment which contains the real storage location of the first word in the page. Figure 3 depicts the new structure. The segment table origin register points to the beginning of the segment table. The segment table in turn points to the page table beginnings. Page tables contain the real starting address of each page. The actual address is then the concatenation of the page frame address and the referenced displacement within the page.

Associative Array Registers

The CPU's DAT feature must reference the segment and page tables which reside in real storage. Real storage speed is slow compared to the speed of the CPU. To maintain the storage management improvements gained by segmentation and paging without a significant reduction in execution speed some special hardware, called associative array registers, is implemented. These registers are faster than memory and compensate for the relatively slow speed of table translation. All System/370 models which can support virtual stor-

RELATIVE ADDRESS

SEG. NO.	PAGE NO.	DISPLACEMENT

SEGMENT TABLE
ORIGIN REGISTER

68,000

SEGMENT TABLE

68,000 →

SEG. NO.	PAGE TABLE LOCATION
1	72,000
2	73,000

PAGE TABLE FOR SEG 1

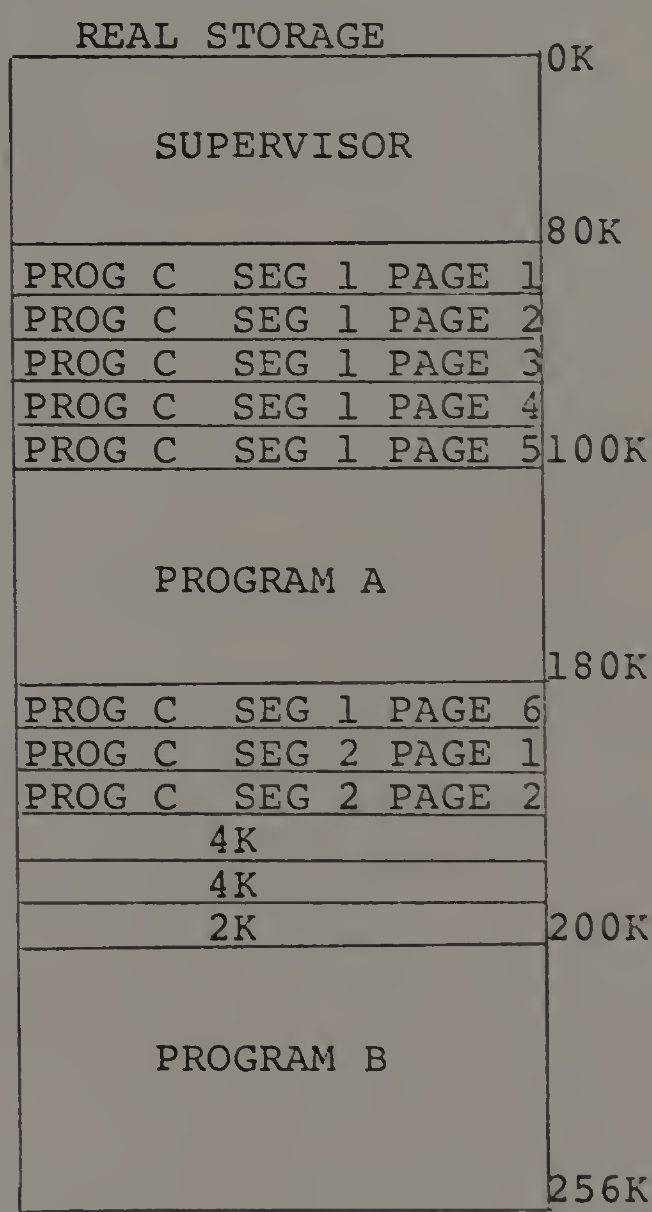
72,000 →

PAGE NO.	REAL STORAGE LOCATION
1	80,000
2	84,000
3	88,000
4	92,000
5	96,000
6	180,000

PAGE TABLE FOR SEG 2

73,000 →

PAGE NO.	REAL STORAGE LOCATION
1	184,000
2	188,000



SEGMENTATION WITH PAGING

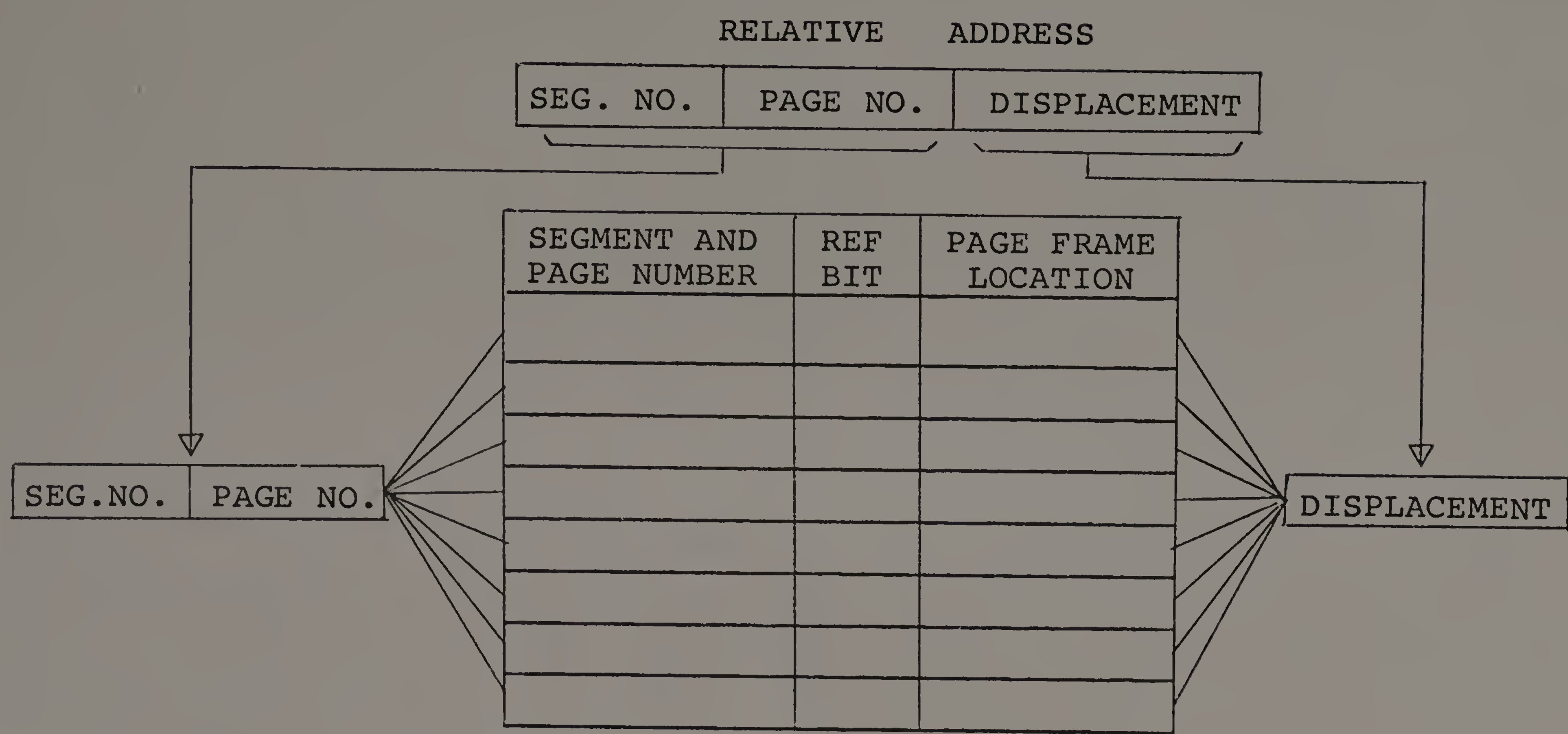
FIGURE 3

age have at least eight such registers. Large models have a special device called a translation look-aside buffer which replaces these registers.

The associative array register is depicted in Figure 4. The eight most recently referenced pages from the executing programs are contained in these registers. The register keeps track of the segment and page number, whether or not the page was recently used (by means of a reference bit) and the actual page frame location. When a relative address translation is requested by an executing program, translation begins simultaneously through the segment and page tables and the associative array registers. The registers are queried in parallel with the desired segment and page number. If a match exists, the page frame location is linked (concatenated) to the relative address displacement to result in the required real storage address. The reference bit of the register is also turned on.

If no match occurs translation will continue through the segment and page tables. The resulting page frame location and its related segment and page number is placed in one of the associative registers having a zero reference bit. The reference bit status is maintained by a hardware implemented Least Recently Used (LRU) algorithm.

Because the associative array registers identify the eight most recently referenced pages in an executing program, they map the most recently referenced 32K in the pro-



RELATIVE ADDRESS TRANSLATION

FIGURE 4

gram assuming 4K pages. Since most address translation occurs through the associative array registers and not the tables, address translation is faster than if it were all done in memory via the tables.

Page Frame Table

Address translation is accomplished by the segment and page tables. Another table called the Page Frame Table performs the memory management function. It keeps track of the status of every page frame in real memory (see Figure 5). If a page is 4K and real storage is 256K, 64 pages will exist in the page frame table. Each of the 64 pages in memory is identified as to (1) the ID of the program currently using the page frame, (2) the segment and page number which specifically identifies the page contained in the page frame, and (3) the status of the page, i.e. whether it is currently being used or is available. The table is used to record the allocation of page frames of real storage to user programs.

Benefits of Segmentation and Paging

In what has been discussed thus far, a paging system alone would be as effective as a segmentation and paging system. Of what use then is segmentation? In short, segmentation provides another level of security protection to the operating software and allows for the provision of another level of storage management. Since all storage references by a program go through the segment tables for trans-

PAGE FRAME NO.	PROGRAM ID	SEGMENT NO. AND PAGE NO.	STATUS
1			0
2			0
3			0
4			0
5			0
.			.
.			.
.			.
.			.
.			.
.			.
.			.
62			0
63			0
64			0

PAGE FRAME TABLE

FIGURE 5

lation, a protection mechanism can be built into the segment that is protected, during address translation the system can generate an interrupt and prevent the illegal reference. If the use of storage protect keys is also used, a hierarchy of protection results. Segmentation also allows a segment's page tables and pages to reside on a direct access storage device (DASD) to be brought into memory only when required. Thus a segment of a program may be executing in main storage while some unneeded segment resides in auxiliary storage until it is demanded. This concept is known as demand paging and is the cornerstone of virtual memory management.

VIRTUAL STORAGE

Memory Size

A computer's addressable memory is limited only to the number of different locations which can be described by the address word used. For example, if a 12-bit address word is used, 4096 unique addresses may be identified. If a 24-bit address were used as is the case in IBM System 360 and System 370 machines 16,777,216 locations could be identified. [3] If a computer's real memory is smaller than 16 megabytes, then 16 megabytes may still be referenced if virtual memory is allowed to exist on some auxiliary storage device and a segmentation and demand paging system is implemented. This system functions to map 16 megabytes of auxiliary storage onto less than 16 megabytes of real storage. Real storage then becomes a resource managed by the operating system.

Several advantages obtain. Since only a small portion of a given program is ordinarily operative at a time, unnecessary pages may reside in secondary storage freeing primary storage to be allocated among more programs according to their actual needs. The following will describe a two-level virtual storage system (OS/370 VS) that uses segmentation and paging.

Implementation of Virtual Storage

There are two methods of implementing virtual storage with multiprogramming. Method one defines the entire system as 16 megabytes of virtual storage. All active programs in the system, including the system control program, are mapped into the single virtual storage. The virtual address space is allocated to programs in segment size increments. Thus the supervisor and all active programs are structured in virtual storage just as they would be structured in real storage in a static relocation system like OS/360. (OS/370 VS is organized in this manner.) A second method defines multiple virtual storage systems in which each program or each user has their own 16 megabytes of address space. Examples are the CP/67 and TSS/360 operating system for the System/360 Model 67 or VM/370 for System/370. Here, the single virtual storage system is described.

In implementing the system, first, the virtual storage which resides on some auxiliary storage device is partitioned into fixed size segments which in turn are divided into fixed size pages. Pages are said to reside in slots of external page storage. All active slots are referenced by an External Page Table which is of the same form as the page table, except the page table maps pages onto primary storage and the external page table maps pages onto virtual storage (see Figure 6). In the event a segment does not have all its pages filled its slots remain unused. Thus virtual (external) storage is fragmented rather than real (primary storage).

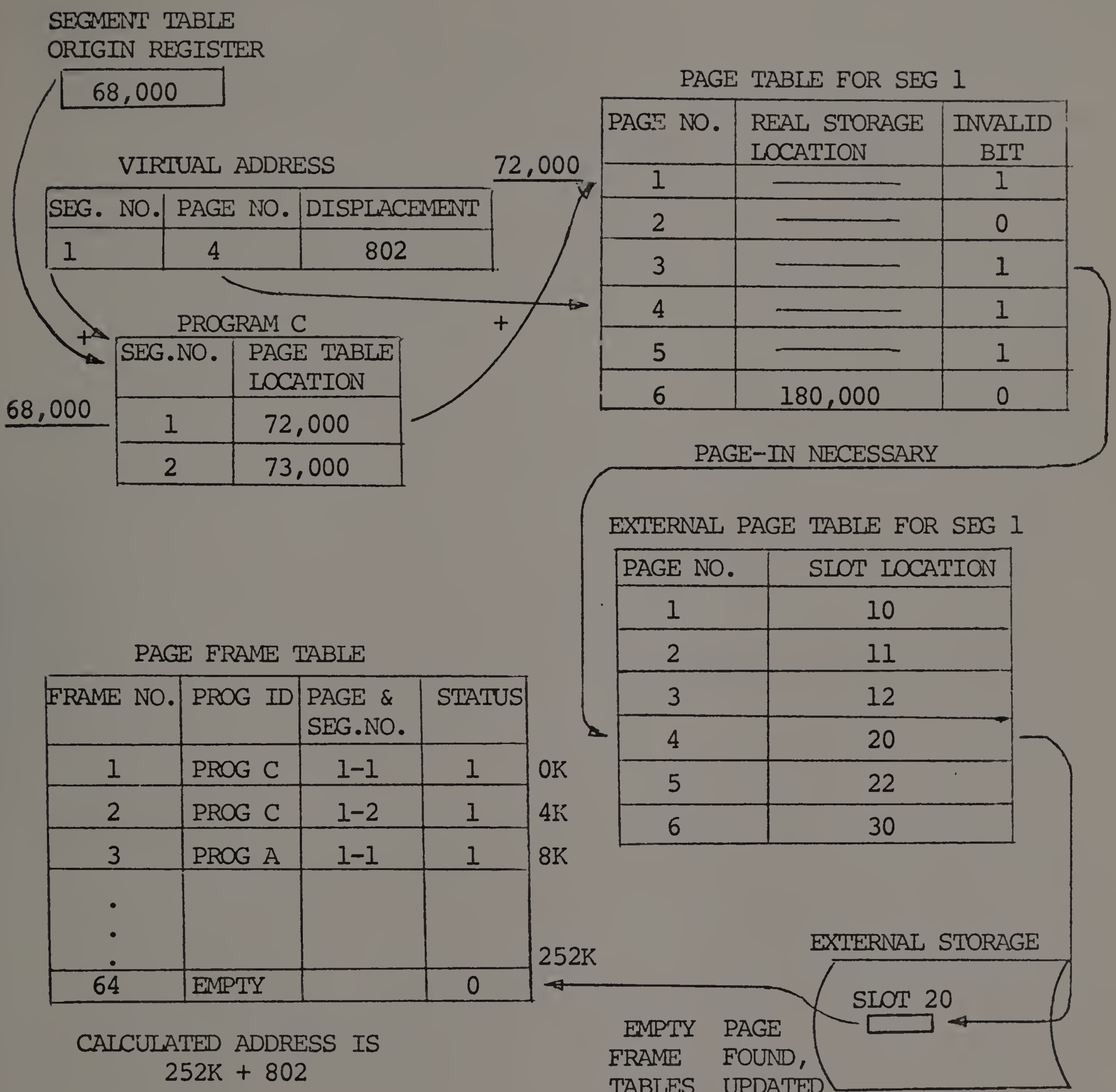
In Figure 6 the external page table is the only additional table required to make the Segmentation and Paging System of figures 3 and 5 a virtual memory system. An additional augmentation is an invalid bit indicator in the page table to denote a page fault, i.e. that the page does not reside in real storage and must be paged in if referenced.

Operation of Virtual Storage

As programs are queued for execution they are segmented, paged and compiled under the direction of the operating system and then placed in external virtual storage as room permits. If the system uses 4K pages, the first eight bits of the 24 bit virtual address refer to the segment number, the next 4 bits the page number and the last 12 bits the displacement within the page. This results in 64K segments and 4K pages with 16 pages in each segment. (VS2 which replaces MVT uses 4K pages.)

Once the program is loaded into virtual storage the program may begin execution. One or more of the program's pages are loaded into available page frames, the page frame table is updated, the appropriate page table entries and their invalid bits are updated, and execution begins. Virtual addresses are translated using the DAT feature and the associative array registers. If a page fault occurs a page-in is demanded (see Figure 6). The system finds the external page address in the corresponding entry of the external page table. The page frame table is searched for a free page. If an open page isfounnd, the page-in operation is executed. The page frame table is updated; the page table is updated; and the page's real storage page frame location is placed into an associative array register. If an empty page does not exist in real memory the Least Recently Used (LRU) rule is applied to the pages. The LRU page is placed in external memory and the referenced page is paged in.

The LRU strategy was selected because if a highly referenced page were selected, the probability that it would be needed again soon, would be high and unnecessary overhead would be incurred through page-in, page-out, page-in. Secondly, if a page is replaced whose contents have changed, the entire page must be written onto external storage instead of only updating the pointers in the tables. The slot selected in page-out need not be the one that contains the old copy of the page, as the system only needs to update the external



VIRTUAL STORAGE QUERY REQUIRING PAGE-IN

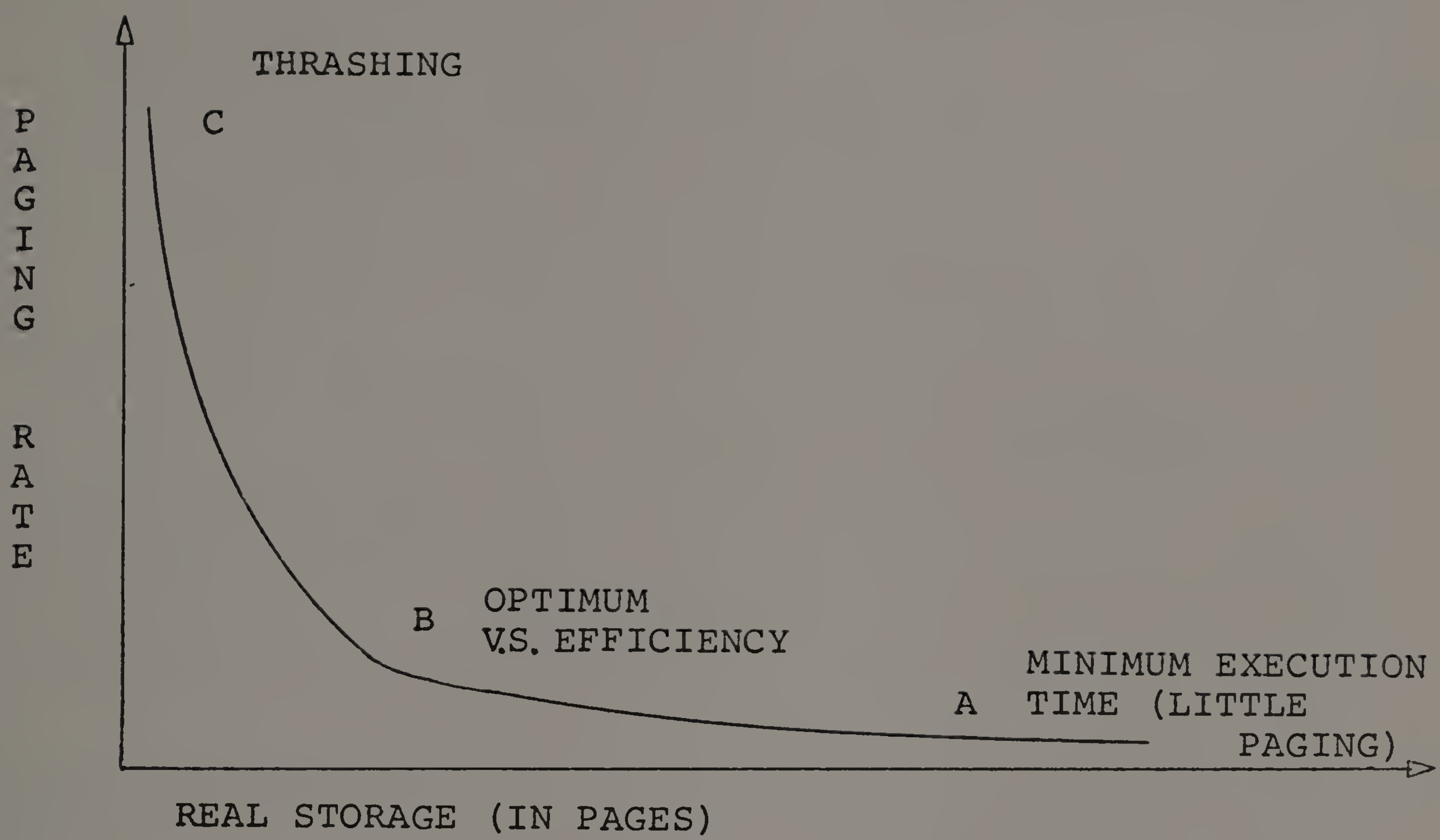
FIGURE 6

page table entry to designate the new slot location. The reference bit and the change bit in the page frame table monitor page usage and page modification. In reality the LRU rule is implemented by replacing the page frame table with a set of page frame queues, the lowest priority queues containing the least recently used pages.

Thrashing

Some studies compare the paging rate with the amount of real storage in pages given a constant load. The result is a negative exponential efficiency curve as depicted in Figure 7. Point A of the curve indicates a low paging rate can be obtained if real storage is large. The condition produced is that the pages are never paged out once they are loaded until program execution terminates. This gives a minimum program execution time since there are no paging delays during program execution. The region of point B is optimal. Real memory is sufficient to produce paging rates having little affect on job load program execution times. Point C indicates a high paging rate, the result of insufficient real storage to support the load. The consequence is long program execution time as pages are continuously swapped from external storage to real storage to external storage. This churning is called thrashing.

Since memory size is a constant but the load is variable, the operation of a virtual storage system can be described as movement along a family of such efficiency curves



PAGING RATE VS. REAL STORAGE SIZE GIVEN
A CONSTANT LOAD

FIGURE 7

as load varies. The control criteria to prevent thrashing is to adjust the system load so as to remain at the point B region. A thrashing monitor achieves this control by sensing paging rates and temporarily halting low priority jobs when the thrashing limit is sensed. The page frames used for these jobs are then freed and applied to the job load to reduce paging. If the paging rate is still too high, another job is interrupted until the paging rate reaches an acceptable limit. In this way the system dynamically adjusts the allocation of resources among users to assure satisfactory performance.

Efficiency of Virtual Storage Systems

Since paging is an I/O operation the CPU is kept busy by executing other programs when reference is made by a program to an element not in real storage. With demand paging if elements on a page are not referenced the page never enters real storage. In general, what code is executed depends entirely on the reference pattern of a program which varies with the data input. Some exception routines are never executed. Normally, during a program's execution its reference pattern will dwell on a relatively small number of pages for a relatively long period of time. This core of pages is called the program's working set.

A program's working set is difficult to predict without monitoring the program over time with various types of transactions. In the case of heavily used programs this may be advantageous. The working set may then be optimized. However, several general suggestions may be implemented in all program designs which increase the locality of reference thus decreasing the probability that paging will be required (see the following section, "Programming Style for Virtual Memory Systems"). Increased locality of reference reduces a program's working set with a resultant decrease in paging operations and the added advantage that now more programs may reside in real storage at a given time before thrashing will occur. The result is an increase in system throughput and effective real memory with increased efficiency of usage of real storage and CPU time. This is one of the big reasons virtual storage systems work to a user's advantage.

Virtual Storage not a Panacea

Virtual storage, however, is not for everyone. Kurtz and Cuozzo of Booz, Allen and Hamilton identify the users who would not benefit from virtual storage. [4] They indicate that users who have a known load and have already optimized operations might do better to increase real storage rather than adopt Virtual Storage. However, ancilliary benefits to such a user might be that of implementing changes without having to worry about reprogramming overlay structures or increasing system size. Bergstresser of IBM, points out that if an installation has several programs that use as much of a partition that is available in order to achieve best performance they could cause abnormally high paging. [5]

Kurtz and Cuozzo point out that users with heavy data base real-time processing, causing high main storage utilization, will achieve the highest gain. [6] Berstresser agrees, pointing out that under VS a high priority job can have more real storage allocated to it during its peak loads than would have been practicable in previous operating systems which is extremely significant for teleprocessing and data base applications. [7] Nothing is free, however, and Data Administrators should not accept claims, but develop and execute a comprehensive planning and analysis program to weigh virtual upgrade alternatives. Central to the analysis is the use of system monitoring and measurement devices to analyze the real benefits of virtual storage. The sections which follow outline the system cost of virtual storage and some areas in which available software may need to be changed to maintain efficiency under the new system.

EFFICIENCY OF VIRTUAL STORAGE

More systems resources are needed to handle the functions of virtual storage than are required by a conventional multiprogramming environment. It is in balancing these requirements with the additional functions and benefits offered by virtual storage in the environment of a system load that a go or no-go decision can be reached. Donald Stroud, Manager of Data Processing and Systems, the Tremco Manufacturing Company, Cleveland, Ohio, complains in a letter to Datamation that someone ought to objectively evaluate and compare hardware and software, and thus provide users with some objective information that will let them decide for themselves what is best for their particular situation. [8] Datamation answers that objective information is not always available, that metrics is a dark art staggering toward the Hall of Science, and that the ACM Special Interest Group on Measurement and Evaluation (SIGME) is working at the frontiers. To this an independent computer measurement consultant and a member of SIGME states that he is certain that no comprehensive analysis effort is being undertaken by any group. [9] As concluded in Chapter VI, "Measuring and Analyzing Computer System and Data Base Effectiveness," computer hardware and software systems cannot be analyzed in general simply because no one is certain what variables in general describe system performance.

The analysis must then take on a general comparative flavor. For example, it can be said that resident control program space is increased by requirements for virtual storage management routines; additional CPU time is required for such functions as dynamic address translation, processing of paging interrupts and the translation of channel program virtual addresses; and I/O time is needed for paging operations.

Memory requirements for control programs is not as significant in a large system where the space needed for the control program can be recovered by better storage management. As real storage becomes smaller, however, storage overhead becomes more significant. As an indication of additional memory requirements, a VS1 system can be compared to its MFT predecessor. A stripped VS1 system might require 64K, while the MFT would require a 40K nucleus. Since the subsystems are unchanged, the 24K would be most of the difference between VS1 and MFT residency. [10]

The amount of real storage required by the resident part of the operating system detracts from the amount of storage available for paging. The fixed real storage, however, varies with processing because the amount of memory required for tables depends upon the amount of virtual storage in use, i.e. system load, as does the amount of external page storage required. Also varying space is short term fixed page space necessary to handle I/O activity.

The paging rate is directly affected by the ratio of utilized virtual storage to the amount of pageable real storage, and the percentage of active data and program information within this utilized space. The result is the efficiency curve of Figure 7. A high inception of the thrashing monitor control program may be indicative of the need for more real storage or a higher speed direct access storage device. To monitor this activity virtual storage operating systems output paging statistics. VS1 and VS2 provide both job and system-wide page-in, page-out statistics while DOS/VS provides a page fault trace capability.

Virtual storage systems also utilize at least part of a channel's time for page transfers between primary (real) and auxiliary (secondary) storage. Device speed, however, is only one factor to be considered here. The control program's ability to spread external page storage over multiple devices in order to permit overlapping of paging activity is at least as important. Often, performance can also be increased by reducing the other I/O activity on the channel to which the paging devices are attached.

These system costs and considerations must be contrasted with several system benefits which accrue to the user of virtual storage. The most important consideration is that the system can initiate and execute more jobs since only their working set exist in real memory. Also dividing programs into pages significantly reduces fragmentation of real memory. Substituted is fragmented auxiliary storage, the result of segmentation. Since auxiliary storage is cheaper than primary memory and cannot access the CPU to execute job steps anyway, the effect is an increase in real memory usage and system throughput.

Programmers experience efficiencies also. Overlay programming is eliminated making system change more flexible, simpler and less costly. The implementation of new systems is also simplified. They can be tested in a low priority partition without significantly affecting the working load. Also long running applications will use only the amount of real storage required at any point in time. The augmentation of time sharing operations and on-line data entry is thus simplified and can be allowed to grow at its own rate without reprogramming the operating system. Advantages accrue to operating systems themselves in that infrequently used routines may be paged. Their flexibility is increased without eating up real storage.

Finally, an amount of system protection has been added. A smaller system may now serve as back-up for a larger system. In short virtual storage system costs may be significant to the small user or the user who has already optimized his system. However, it is a boon to the implementation of the computerized data base concept within organizations as it provides a new dimension to total system flexibility.

PROGRAMMING STYLE FOR VIRTUAL STORAGE

Though virtual storage provides huge systems benefits and flexibility and increased efficiency to larger users it also places constraints on programming style which had not previously existed or whose consequences were less severe. Programming with overlays in mind or with excessive branches to distant parts of the program are out. However, users that have modularly structured programs still have the key to efficient execution, under a virtual storage or a non-virtual storage operating system. [11] Programs with disciplined flow and data organization, those written to conserve storage space, are highly effective in a virtual storage environment because they have a relatively small working set.

A program's working set is of course to some extent unpredictable because it depends on the data input. However, some programming techniques can be applied to improve locality of reference, that is, to keep things used with each other in time, close to each other in location. For example, unusual-situation routines can be grouped away from the main program flow. If a short subroutine is used only once or twice it can be included in the calling program to reduce reference to another page. Subroutines should be placed in their most probable succession, so that processing proceeds sequentially, with calls being made to infrequently used subroutines. Infrequently used subroutines should be located together as well to increase the probability that they will be placed within the same page.

In addition the data used by a routine should be placed near that subroutine instead of scattering it over the program and many pages. In particular data should not be initialized at the beginning of the program and then used much later on as this will increase the probability of data change occurrence. In the page-out operations, a new page is rewritten on secondary storage only if it has changed in content. Otherwise, only the program's page table is updated. This also is the reason that self-modifying code should be reduced or eliminated.

Other suggestions concern placing data so that it can be accessed sequentially in memory. Thus data should be referenced in the order in which it was stored. Hence, chains are bad because they must be searched.

Each of the above suggestions reduce the number of page faults with their resulting paging activity. In retrospect one may conclude that modular programming is still a good technique, but is works best in virtual storage if not carried to an extremely low level.

VIRTUAL STORAGE NOT A NEW CONCEPT

Virtual storage is not a new concept. [12] It was used fifteen years ago in the Atlas and Gamma 60, and was implemented on the B5000 machines in 1962. Univac followed in 1963 with CDC and DEC in 1965. Gaudion, tongue-in-cheek points out that IBM was the first to coin the catch phrase, "Virtual Memory," and the whole thing is an IBM marketing ploy to sell the world someone else's invention. Well, maybe, but virtual storage adds a degree of flexibility to processing of information which will allow development of computer management information systems to new higher levels. Software is made independent of main storage; system software may be made larger to optimize performance and be more comprehensive; development of applications are speeded, they can be debugged in low priority partitions or on smaller machines; programs whose storage requirements vary widely during execution, based on transaction volume and complexity, can be justified, designed, installed and run more easily; and finally virtual storage allows applications to be run on machines of widely varying size. Virtual storage has been with us for some time, it has been reintroduced now to meet a need pressed by this generation of management information systems.

FOOTNOTES

1. *Introduction to Virtual Storage in System/370*, IBM Corporation, GR20-4260-0, August 1972.

2. J.F. Kurtz and D.E. Cuozzo, "How to Get Benefits From Virtual Storage," *Datamation* (February 1973), p. 50.

3. The unique addresses expressible by a 12-bit word can be calculated in the following manner: The largest number expressible in 12 bits is $2^{12} - 1$. The thirteenth bit represents 2^{12}. It is turned on if a one is added to a 12 position binary number which is all ones. $2^{12} - 1$ is 4096 - 1 or 4095, but the number zero must be included for a total of 4096 possible addressable locations using a 12 bit address.

4. J.F. Kurtz and D.E. Cuozzo, "How to Get Benefits From Virtual Storage," *Datamation* (February 1973), p. 48-52.

5. R.V. Bergstresser, "Virtual Storage Operation," *Datamation* (February 1973), p. 57.

6. *Ibid*., p. 51.

7. *Ibid*., p. 56.

8. "Virtual Measurement," Letters, *Datamation* (December 1972), p. 25.

9. "IBM's Gentle Nudge," Letters, *Datamation* (February 1973), p. 24.

10. R.A. McLaughlin, "IBM's Virtual Memory 370's," *Datamation* (September 1972), p. 60.

11. J.F. Kurtz and D.E. Cuozzo, "How to Get Real Benefits From Virtual Storage," *Datamation* (February 1973), p. 52.

12. E.J. Gaudion, "Today IBM Announces Yesterday," *Datamation* (February 1973), p. 53-54.

APPENDIX D

APPENDIX D

ADVANCES IN MEMORY TECHNOLOGY

MAGNETIC BUBBLE MEMORY

LSI and Bubble Memory Fill Gap in Memory Hierarchy

LSI technology and magnetic bubble technology are especially promising because they fill a time gap factor of 10^4 existing between access times of electromechanical and core memory in the storage hierarchy. Magnetic bubble memories are easier to construct than LSI memory because all elements are passive and no diffusion is required. In addition bubble memory is nonvolatile, that is, memory content is not affected by voltage fluctuations or power shut-off.

In a 1971 International Magnetics Conference held in Denver, April 13-16, Jack Morton, Vice President, Bell Telephone Labs contrasted the developing LSI technology and magnetic bubble memories with magnetic core, disks and drums. [1] He sees core memories bottoming at one microsecond cycle times, a cost of one cent per bit and power dissipation of 100 microwatts per bit in greater than megabit sizes. Also, the constraint of minimum module size imposed by cores, disks and drums may severely limit computer architecture. LSI memories, Morton predicts, will reach costs below one tenth of a cent per bit at cycle times of 10^{-8} seconds and power dissipation of 10 microwatts per bit.

Magnetic bubbles are promising because their power dissipation is similar to LSI memories, but their nonvolatility like that of disk memory or core memory assures that loss of data will not occur during power transients or power loss. Another important plus of LSI and bubble memory is that their content is directly addressable by the CPU. It is possible to perform some of the logic with bubble technology, resulting in an intimate mix between memory and logic. However, tolerances for logic are much tighter than memories, and the interaction among bubbles necessary to construct AND and OR gates leads to complexity. Therefore, logic will probably be used only within memories, where logic requirements are modest and it is desireable to build storage devices with a single type of memory. [2]

The use of bubble memory in business systems lies in data storage and high volume data storage rather than replacing computer LSI logic. According to William Mavity of North American Rockwell's Electronics Group, their importance to computer architecture in the mid and late 70's is their ability to operate under the control of external clocking, the capability to operate at any speed, stop for any length of time and reverse direction of data flow, coupled with significant reductions in data access time. [3]

Bubble Memory Realization

Recent developments in magnetic bubble technology realization occurred when Praveen Chaudhari of IBM demonstrated that magnetic bubbles could be formed in amorphous films (material in which the molecules are not set in rigid order). [4] Previously, all research was being conducted with crystalline film (materials in which the molecules form a lattice). Crystalline film requires that first a perfect crystal be grown under a complex high temperature process, then the cyrstal is polished smoother than a gem before the film is deposited. In contrast, amorphous film can be deposited on ordinary window glass at room temperature. [5] The cost reduction possibilities are amazing. Hogan of Fairchild semiconductor once said that (crystal-based) magnetic bubbles were dead as far as commercial memories are concerned but now recants that, "Every time there's a major breakthrough (amorphous magnetic bubbles), I change my mind." [6]

Another product worth mentioning is the work being done at Cambridge Memories, Inc. in Newton, Massachusetts. Cambridge uses a polycrystalline film instead of the amorphous material. [7] Deliveries of this memory for use in intelligent terminals and other commercial applications are occurring now. Though memories are being turned out at the modest rate of one million bits per day, a higher capacity production unit in Spain will be operating by May 1973. [8]

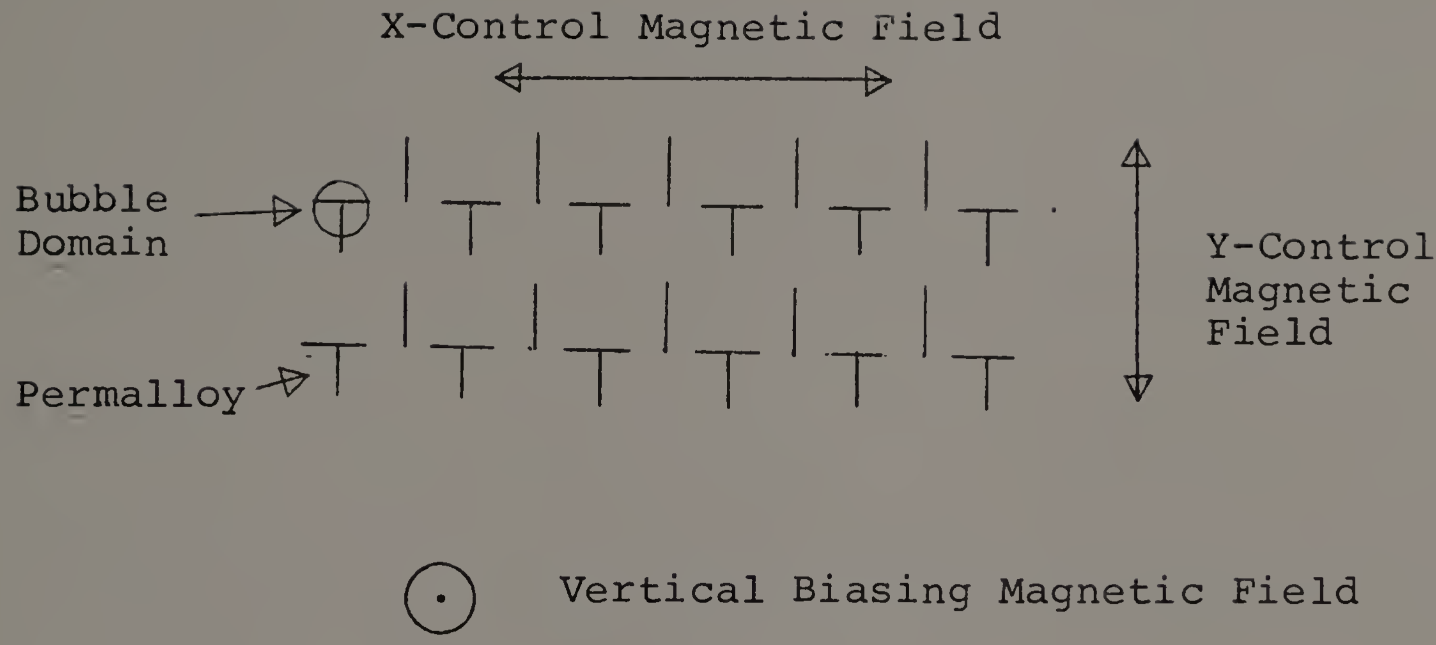
The memories are manufactured by evaporating aluminum onto glass, etching it to form grooves to steer the magnetic spots and then covering the aluminum with polycrystalline nickel-iron-cobalt magnetic material. Memories will cost less than half that of core and about the same price as large disk units in competing data volume, but offer more reliability than mechanical disks. [9] Memories of this construction can never meet the volume/speed possibilities of magnetic bubbles and therefore are only an interim product.

A competing technology, charge-coupled devices are the electrical analog of bubble memories, and in fact were constructed after bubble memory research commenced. A negative aspect of charge-coupled devices is volatility, they do not retain data when power is disconnected. For a more complex description of charge-coupled devices see Boyle and Smith, IEEE Spectrum, July 1971. [10]

Magnetic Bubble Technology

Magnetic bubble storage has the potential to provide access times of tens or hundredths of a microsecond with data rates to ten billion bits per second. [11] It has long been known that under a strong magnetic field magnetic domains can be formed. Magnetic bubble memory research has only commenced in the mid 1960's. Basically, magnetic bubbles are formed by inducing a magnetic field perpendicular to a crystalline or amorphous film. The magnetic field causes the formation of a cylindrical domain, appearing as a mountain on the film, which can be seen with a microscope. These domains can be moved about in the film by bias fields applied horizontally and laterally. Information is stored as the presence or absence of a domain or bubble at a specific location.

To control the flow of bubbles current carrying loops are deposited horizontally and vertically forming a matrix of criss-crossing wires. The wires resemble channels as a current carrying wire lies on each side of the bubble path. A basic modification of the dual wire channel is the currently popular T-bar method. Rather than the intersecting channels controlling bubble position and movement, a series of alternating rows of deposited metal T's and I's act in a rotating field, pushing and pulling magnetic bubbles along from T to I to T (see Figure 1). The rotational speed of the magnetic field determines how fast the bubbles move. An anology can be drawn with an electric motor. In an electric motor the frequency of excitation (and the coil configuration) determine motor speed as the rotating magnetic field pushes and pulls against stator protrusions. Lenz's law is active here, it says that wherever there is a magnetic field in motion, there is current flow induced in any closed circuit; and wherever there is current, there is a magnetic field. Only in magnetic bubble technology, the magnetic field moves the bubbles of magnetism induced by a bias field, and in electric motors the rotor turns in an attempt to reach



MAGNETIC BUBBLE T-BAR-CONTROL DESIGN

FIGURE 1

an equilibrium which never exists until the excitation is removed. A more technical discussion of magnetic bubble memory appears in the September, 1972 issue of the IEEE Spectrum. [12]

Applications thus far have concentrated on the serial movement of bubbles in shift-registers and disk storage. [13] However, flip-flops and logic gates are also possible if two or more data streams interact, though tolerance for logic designs are more stringent than for memories. [14] Currently, Bell labs has completed a 16 million-bit bubble memory, and IBM is operating a 1000 bit shift register. [15] Technological possibilities run to .1 micron bubbles yielding densities of 1 billion bits per square inch. [16]

Magnetic bubbles are easier to make than LSI devices because no diffusion is required, Permalloy and conductive layers need only be deposited and etched. A final metalizatión process defines interconnects, current-carrying functions, and interconnection pads. Conventional integrated circuits provide sensing and control functions.

Sensing is Hall-effect or magnetoresistive. The Hall-effect sensor affects a semiconductor's current flow in such a way that a voltage is produced at right angles to the semiconductor current flow. Magnetoresistive sensors work on the principle that the resistance of the material (usually Permalloy) changes when a magnetic field is applied perpendicular to resistor current flow. The resistance change causes a change in the voltage across the resistor as the magnetic bubble moves past the magnetoresistive device.

OPTICAL MEMORIES

Optical Memory May Replace Entire Memory Hierarchy

Another technology several years behind Bubble Memory in realization is optical memories. Optical memory organization may be of bit-by-bit serial design or holographic parallel page-by-page input/output. Both systems employ optics, lasers, deflection systems and some storage medium in varying degrees of complexity depending on the particular design. RCA's holographic design, for example, employs a laser, liquid crystals, electro-acoustic deflectors and holograms stored on thermoplastics. Information is stored on thermoplastic holograms formed by scattering light through liquid crystals whose opaqueness or clarity is controlled by electronic signals. Reading is accomplished by scattering a laser beam through the hologram onto a light-sensitive array. Erasure is performed by heating the hologram. [17] The RCA system could store as much data as large disk systems currently in existence, but would be 1000 times faster.

Thomas O. Stanley, staff vice-president of research programs for RCA predicts that holographic memory has the potential when fully developed, to replace the entire hierarchy of core, drum and disk systems now used, and thereby simpligy the whole architecture of computers. [18] This means that there will be no peripherals whose access time must be masked and therefore multiprogramming and virtual storage will be eliminated and MIS file structures simplified. [19]

The State-of-the Art

Optical memory devices are being developed because magnetic disk devices are presently storage limited to densities of 8×10^5 bits per in^2 on a recording area 10^4 in^2. Addressing areas greater than this or increases in bit densities are not currently feasible. Greater storage volume appears to lie with optical memories since they are limited by the diffraction limit of light, approximately .5 micrometers for visible light and f/1 optics. Bit densities of $1.5x10^8$ bits per in^2 have been experimentally demonstrated in the laboratory. Precision Instrument Company has built a system which can store 10^{12} bits at a density of $2x10^7$ bits/in^2. [20]

For a detailed analysis of optical memory state-of-the-art see Tuft and Chen, IEEE Spectrum, February and March 1973. [21] Some points concerning optical memories are summarized below:

1. The major cost lies in the optics, therefore, cost per bit decreases rapidly for large memories.

2. High bit densities can be achieved without positioning the final objective lens extremely close to the recording medium. By contrast magnetic disk recording head-recording media distance must be comparable to the bit density, increasing the possibility of head crashes.

3. The system is inertia-free, having no moving parts (bit-by-bit systems require some movement).

4. Readout signal intensity is proportional to laser beam power for optics, whereas readout signal intensity depends on bit size for magnetic disk memories.

5. Page organized format allows parallel input/output of data. Access time is less than 10 microseconds, three orders of magnitude better than the magnetic disk capability.

6. Holographic techniques are tolerant of dust or defects in the storage medium, since the hologram is of the Fourier transform of the bit pattern.

FOOTNOTES

1. "Intermag-1971," Datamation (September 15, 1971), p. 37.

2. G. Lapidus, "The Domain of Magnetic Bubbles," IEEE Spectrum (September, 1972), p. 62.

3. E.D. Lundell, "Magnetic Bubbles Place Seen Between Core and Disks," Computerworld (September 27, 1972), p. 24.

4. "Why IBM Got a Jump in Bubble Memories," Business Week (February 17, 1973), p. 38.

5. Ibid., p. 40.

6. Ibid.

7. "IBM Bubbles May Be in User's Future," Computerworld (February 28, 1973), p. 18.

8. "The Poor Man's Magnetic Bubble Memory," Business Week (April 29, 1972), p. 80.

9. Ibid.

10. W.S. Boyle and G.E. Smith, "Charge-Coupled Devices--A New Approach to MIS Device Structures," IEEE Spectrum (July, 1971), pp. 18-27.

11. Lapidus, Op. Cit., p. 58.

12. Lapidus, Op. Cit., p. 58-62.

13. A.H. Bobeck and H.E.D. Scovil, "Magnetic Bubbles," Scientific American (June, 1971), p. 79.

14. Lapidus, Op. Cit., p. 62.

15. "Why IBM Got the Jump in Bubble Memories," Business Week (February 17, 1973), p. 38.

16. Ibid., p. 40.

17. "RCA Uses Holographic Approach, IBM Puts 8K on a Chip," Computerworld (February 28, 1973), p. 43.

18. Ibid., p. 43.

19. J.G. Roberts, "Asymetric Memory Hierarchies," ACM Communications, Vol. 16, No. 4 (April, 1973), pp. 214-215.

20. O.N. Tufte and D. Chen, "Optical Techniques for Data Storage," IEEE Spectrum (February, 1973), p. 26.

21. O.N. Tufte and D. Chen, "Optical Techniques for Data Storage," IEEE Spectrum (February, 1973), pp. 26-32 and O.N. Tufte and D. Chen, "Optical Memories: Controlling the Beam," IEEE Spectrum (March, 1973), pp. 48-53.

BIBLIOGRAPHY

"A Surprising Potential for New Gains." Business Week, September 9, 1972, pp. 88-94.

Aaron, J.D. "Information Systems in Prespective." Computing Surveys, Vol. 1, No. 4, December, 1969, pp. 213-236.

"Accuracy is Intelligent Terminals' Chief Asset." Computerworld, February 28, 1973, p. 4.

Ackoff, R.L. "Management MisInformation Systems." Management Science, Vol. 14, No. 4, 1967, B147-B156.

Anacker, W. and Wange, C.P. "Performance Evaluation of Computing Systems with Memory Hierarchies." IEEE Transactions on Electronic Computers, Vol. EC-16, No. 6, December, 1967, pp. 764-773.

Anderson, H.A. Jr. and Sargent, R.G. "Bibliography 31, Modeling, Evaluation, and Performance Measurements of Time-Sharing Computer Systems." Computing Reviews, December, 1972, pp. 603-608.

Anderson, R.H. and Sibley, W.L. A New Approach to Programming Man-Machine Interfaces. Santa Monica: Rand, R-876-ARPA, March, 1972.

Anthony, R.N. Planning and Control Systems: A Framework for Analysis. Boston: Graduate School of Business Administration, Harvard University, 1965.

Apple, C.T. "The Program Monitor--A Device for Program Performance Measurement." Proceedings, ACM 20th National Conference, 1965, pp. 66-72.

Arbuckle, R.A. "Computer Analysis and Thruput Evaluation." Computers and Automation, January, 1966, pp. 12-19.

Aronson, R.L. "Chevrolet Saginaw--A Big Control System Starts Up." Control Engineering, May, 1972, pp. 60-65.

Ashby, W.R. Design for a Brain. New York: John Wiley & Sons, Inc., 1960.

Ashby, W.R. An Introduction to Cybernetics. Science Editions. New York: John Wiley & Sons, Inc., 1963.

Bachman, C.W. "The Evolution of Storage Structures." Communications of the ACM, Vol. 15, No. 7, July, 1972, pp. 628-636.

Baer, J.L. "A Survey of Some Theoretical Aspects of Multiprocessing." Computing Surveys, Vol. 5, No. 1, March, 1973, pp. 31-80.

Bairstow, J.N. "A Review of Systems Evaluation Packages." Computer Decisions, June 1970, p. 20.

Bairstow, J.N. "The Terminal that Thinks for Itself." Computer Decisions, January, 1973, pp. 10-13.

Balzer, R.M. On the Future of Computer Program Specification and Organization. Santa Monica, Calif.: Rand Corp., R-622-Arpa, August, 1971.

"Banking System Can Replace Tellers." Computerworld, February 28, 1973, p. 21.

Baran, P. On Distributed Communications: IX. Security, Secrecy, and Tamper-Free Considerations. Santa Monica, Calif.: Rand Corp., RM-3765-PR, August, 1964.

Bard, Y. "Performance Criteria and Measurement for a Time-Sharing System." IBM Systems Journal, Vol. 10, No. 3, 1971, pp. 193-216.

Beardsley, C.W. "Is Your Computer Insecure?" IEEE Spectrum, January, 1972, pp. 67-78.

Becker, H.B. "Information Network Design Can Be Simplified Step-By-Step," Computer Decisions, October, 1972, pp. 14-17.

Beckmann, P. "Computerization of English." IEEE Spectrum, December, 1971, pp. 20-27.

Beer, Stafford. Cybernetics and Management. Science Editions. New York: John Wiley & Sons, Inc., 1959.

Beer, Stafford. Decision and Control. New York, John Wiley & Sons, Inc., 1966.

Beer, Stafford. Management Science. London: Aldus Books Ltd., 1967.

Bell. C.G. and Newell, A. "Computer Structure--Past, Present and Future." Fall Joint Computer Conference Proceedings, 1971, pp. 387-396.

Bell, D.A. Intelligent Machines. Blaisdell Publishing Company. New York: Random House, Inc., 1962.

Bell, T.E. "Choose Your Tools to Check Your Computer." Computer Decisions, November, 1972, pp. 12-15.

Bell, T.E. Computer Performance Analysis: Measurement Objectives and Tools. Santa Monica, Calif.: Rand Corp., R-584-NASA/PR, February 1971.

Bell, T.E. Computer Performance Analysis: Minicomputer-Based Hardware Monitoring. Santa Monica: Rand, R-696-PR, June, 1972.

Bell, T.E. "Objectives and Problems in Simulating Computers." Fall Joint Computer Conference, Vol. 41, 1972, pp. 287-297.

Bell, T.E.; Boehm, B.W. and Watson, R.A. Computer Performance Analysis: Framework and Initial Phases for A Performance Improvement Effort. Santa Monica: Rand Corp., R-549-PR, August, 1971.

Benjamin, R.I. "A Generational Prespective of Information Systems Development." Communications of the ACM, Vol. 15, No. 7, July, 1972, pp. 640-643.

Benjamin, R.I. Control of the Information System Development Cycle. New York: John Wiley & Sons, Inc., 1971.

Benner, Frank K. "On Designing Generalized File Records for Management Information Systems." Fall Joint Computer Conference Proceedings, 1967, pp. 291-303.

Bergstresser, R.V. "Virtual Storage Operation." Datamation, February, 1973, pp. 55-57.

Bernstein, G. "What's in the Cards for Data Entry?" National Computer Conference Proceedings, Vol. 42, 1973, pp. 765-772.

Bertalanffy, Ludwig von. General System Theory. New York: George Braziller, Inc., 1968.

Berztiss, A.T. Data Structures. New York: Academic Press, 1971.

Bisco, R.L., ed. Data Bases, Computers, and the Social Sciences. New York: John Wiley & Sons, Inc., 1970.

Boehm, B.W. Computer Systems Analysis Methodology: Studies in Measuring, Evaluating, and Simulating Computer Systems. Santa Monica: Rand Corporation, R-520-NASA, September, 1970.

Bonner, A.J. "Using System Monitor Output to Improve Performance." IBM Systems Journal, Vol. 8, No. 4, 1969, pp. 290-298.

Bookman, P.G.; Brotman, B.A., and Schmitt, K.L. "Use Measurement Engineering for Better System Performance." Computer Decisions, April, 1972, pp. 28-32.

Booth, G.M. "The Use of Distributed Data Bases in Information Networks." First International Conference on Computer Communications Proceedings, New York: Association for Computing Machinery, October 24-26, 1972, pp. 371-376.

Bowdon, E.K., Sr.; Mamrak, S.A.; and Salz, F.R. "Simulation--A Tool for Performance Evaluation in Network Computers." National Computer Conference Proceedings, Vol. 42, 1973, pp. 121-132.

Boyle, W.S.; Smith, G.E. "Charge-Coupled Devices--A New Approach to MIS Device Structures." IEEE Spectrum, July, 1971, pp. 18-27.

Blumenthal, S.C. Management Information Systems: A Framework for Planning and Development. Englewood Cliffs, N.J.: Prentice-Hall, 1969.

Bracchi, G.; Fedeli, A.; and Paolini, P. "A Relational Data Base Management System." Proceedings of the ACM Annual Conference, August, 1972, pp. 1080-1089.

Bride, E.J. "Brainy Operations Staffs Are an Asset: McFarlan." Computerworld, July 5, 1972, p. 4.

Bride, E.J. "Panel Warned of SS Number Trend." Computerworld, August 30, 1972, p. 1 & 4.

Bride, E.J. "POS Terminals Boost Data Accuracy, Cut Report Time." Computerworld, August 9, 1972, p. 3.

Bride, E.J. "Users Awaken to Security Needs." Computerworld, December 6, 1972, pp. 1 & 4.

Bryan, J.N.F. "Defining The Requirements for A Data Management System." Presented at Guide International, Miami Beach, Florida, May 17, 1972, (Paper may be obtained from CMS Action Group, P.O. Box 2238 AMF, Miami, Fla. 33159).

Bucholz, W. "A Synthetic Job for Measuring System Performance," IBM Systems Journal, Vol. 8, No. 4, 1969, pp. 309-318.

Busignies, H. "Communication Channels." Scientific American, September, 1972, pp. 99-113.

Calingaert, P. "System Performance Evaluation: Survey and Appraisal." Communications of the ACM, Vol. 10, No. 1, January, 1967, pp. 12-18.

Camenzind, H.R. "A Guide to Integrated-Circuit Technology." Electro-Technology, February, 1968, pp. 49-64.

Campbell, D.J. and Heffner, W.J. "Measurement and Analysis of Large Operating Systems During System Development." Fall Joint Computer Conference Proceedings, Vol. 33, pt. 1, 1968, pp. 903-914.

"Canada to Begin All-Digital Dataroute." Computerworld, March 14, 1973, p. 17.

Canning, R.G. "Advanced Projects in Data Processing." EDP Analyzer, Vol. 9, No. 11, November, 1971.

Canning, R.G. "Computer Security: Backup and Recovery Methods." EDP Analyzer, Vol. 10, No. 1, January, 1972.

Canning, R.G. "Creating the Corporate Data Base." EDP Analyzer, Vol. 8, No. 2, February, 1970.

Canning, R.G. "Current Status in Data Communications." EDP Analyzer, Vol. 7, No. 3, March, 1969.

Canning, R.G. "Data Processing Planning Via Simulation." EDP Analyzer, Vol. 6, No. 4, April, 1968.

Canning, R.G. "Data Security in the CDP." EDP Analyzer, Vol. 8, No. 5, May, 1970.

Canning, R.G. "Developments in Data Transmission." EDP Analyzer, Vol. 11, No. 3, March, 1973.

Canning, R.G. "Distributed Intelligence in Data Communications." EDP Analyzer, Vol. 11, No. 2, February, 1973.

Canning, R.G. "Future Trends in Data Communications." EDP Analyzer, Vol. 7, No. 4, April, 1969.

Canning, R.G. "Intelligent Terminals," EDP Analyzer, Vol. 10, No. 4, April, 1972.

Canning, R.G. "New Dimensions in Management Control." EDP Analyzer, Vol. 6, No. 2, February 1968.

Canning, R.G. "Organizing the Corporate Data Base." EDP Analyzer, Vol. 8, No. 3, March, 1970.

Canning, R.G. "Processing the Corporate Data Base." EDP Analyzer, Vol. 8, No. 4, April, 1970.

Canning, R.G. "Progressive Fast Response Systems." EDP Analyzer, Vol. 8, No. 10, October, 1970.

Canning, R.G. "Savings from Performance Monitoring." EDP Analyzer, Vol. 10, No. 9, September, 1972.

Canning, R.G. "Security of the Computer Center," EDP Analyzer, Vol. 9, No. 12, December, 1971.

Canning, R.G. "Systematic Methods for Business Planning." EDP Analyzer, Vol. 6, No. 3, March, 1968.

Canning, R.G. "Technical Support for an MIS." EDP Analyzer, Vol. 7, No. 11, November, 1969.

Canning, R.G. "The 'Data Administrator' Function." EDP Analyzer, Vol. 10, No. 11, November, 1972.

Canning, R.G. "The Debate on Data Base Management." EDP Analyzer, Vol. 10, No. 2, March, 1972.

Canning, R.G. "The Emerging Computer Networks." EDP Analyzer, Vol. 11, No. 1, January, 1973.

Canning, R.G. "The Minicomputer's Quiet Revolution." EDP Analyzer, Vol. 10, No. 12, December, 1972.

Canning, R.G. "Trends in Data Management, Part I." EDP Analyzer, Vol. 9, No. 5, May, 1971.

Canning, R.G. "Trends in Data Management, Part II." EDP Analyzer, Vol. 9, No. 6, June, 1971.

Canning, R.G. "Using Shared Data Files." EDP Analyzer, Vol. 8, No. 11, November, 1970.

Canning, R.G. "What's the Status of MIS?" EDP Analyzer, Vol. 7, No. 10, October, 1969.

Cantrell, H.N. and Ellison, A.C. "Multiprogramming System Performance and Analysis." Spring Joint Computer Conference Proceedings, Vol. 32, 1968, pp. 213-221.

Carlson, G., "How to Save Money with Computer Monitoring," Proceedings of the ACM Annual Conference, 1972, pp. 1018-1023.

Carne, E.B. "Telecommunications: Its Impact on Business." Harvard Business Review, July-August, 1972, pp. 125-132.

Casey, R.G. "Design of Tree Networks for Distributed Data." National Computer Conference Proceedings, Vol. 42, 1973, pp. 251-258.

"Cash Drawers that 'Talk Computer'." Business Week, August 29, 1970, pp. 66-67.

Cheng, P.S. "Trace-Driven System Modeling." IBM Systems Journal, No. 4, 1969, pp. 280-289.

Chervany, N.L. and Dickson, G.W. "An Experimental Evaluation of Information Overload in a Production Environment." Management Information Systems Research Center Working Paper Series, Minneapolis, Minn.: University of Minnesota, 1972.

Churchman, C. West. The Systems Approach. New York: Delacorte Press, 1968.

Clapper, G.L. "Automatic Word Recognition." IEEE Spectrum, August, 1971, pp. 57-69.

Cockrun, J.S. and Crockett, E.D. "Interpreting the Results of a Hardware Systems Monitor." Spring Joint Computer Conference Proceedings, Vol. 38, p. 197.

"Computer Systems Measurement and Evaluation," The Computer Bulletin, February, 1972, p. 100.

"Computer Utilization Rate Improved by Use of New Time Analyzer." Management Adviser, May, 1971, p. 9.

CODASYL Data Base Task Group Report. New York: Association for Computing Machinery, April, 1971.

CODASYL Systems Committee. Feature Analysis of Generalized Data Base Management Systems. New York: Association for Computing Machinery, May, 1971.

Codd, E.F. "A Relational Model of Data for Large Shared Data Banks." Communications of the ACM, Vol. 13, No. 6, June, 1970, pp. 377-387.

Cohen, E. "Symmetric Multi-Mini-Processors: A Better Way to Go?" Computer Decisions, January, 1973, pp. 16-20.

"Communications: A CW Special Report." Computerworld, July 26, 1972, Supplement pages 1-16.

"Communications, or Getting It Together: A CW Special Report." Computerworld, April 25, 1973, Special Report pages 27-47.

Considerations of Data Security in a Computer Environment. IBM Corporation, G520-2169-0, July, 1970.

Conrath, D.W. "Measuring the Computer's Impact on Organizational Structure." First International Conference on Computer Communications Proceedings, New York: Association for Computing Machinery, October 24-26, 1972, pp. 68-73.

Construction Management System Action Group. Data Administration. System Support Group Data Management Project Working Paper, CM5, P.O. Box 2238, AMF, Miami, Fla. 33159, April 21, 1972.

Construction Management System Action Group. Data Management System Requirements. CMS, P.O. Box 2238 AMF, Miami, Fla. 33159, June 23, 1971.

Cooley, William W. and Lohnes, Paul R. Multivariate Data Analysis. New York: John Wiley & Sons, Inc., 1971.

"CRT's Do Have Their Captive Audience." Computerworld, April 25, 1973, p. 57.

Daley, R.C. and Dennis, J.B. "Virtual Memory, Processes, and Sharing in MULTICS." Communications of the ACM, Vol. 11, No. 5, May, 1968, pp. 306-312.

"'DBD' Cuts Across Applications To Check Data Base Duplication." Computerworld, March 7, 1973, p. 10.

Dearden, J. "MIS is a Mirage." Harvard Business Review. January/February, 1972, pp. 90-94.

DeCegama, A. "A Methodology for Computer Model Building." Fall Joint Computer Conference Proceedings, Vol. 41, 1972, pp. 299-310.

DeCegama, A. Performance Optimization of Multiprogramming Systems. Unpublished Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, Pa., April, 1970.

DeFiore, C.E. and Berra, P.B. "A Data Management System Utilizing an Associative Memory." National Computer Conference Proceedings, Vol. 42, 1973, pp. 181-186.

Denning, P.J. "Third Generation Computer Systems." Computing Surveys, Vol. 3, No. 4, December, 1971, pp. 175-216.

Denning, P.J. "Virtual Memory." Computing Surveys, Vol. 2, No. 3, September, 1970, pp. 153-190.

Dennis, S.F. and Smith, M.G. "LSI-Implications for Future Design and Architecture." Spring Joint Computer Conference Proceedings, 1972, pp. 343-351.

Dewan, Edmond M., ed. Cybernetics and the Management of Large Systems. New York: Spartan Books, 1969.

Diethelm, M.A. "A Method of Evaluating Mass Storage Effects on System Performance." National Computer Conference Proceedings, Vol. 42, 1973, pp. 69-74.

Dodd, G.G. "Elements of Data Management Systems." Computing Surveys, Vol. 1, No. 2, June, 1969, pp. 117-133.

Dolotta, T.A. "Functional Specifications for Typewriter-like Time-Sharing Terminals." Computing Surveys, Vol. 2, No. 1, March, 1970, pp. 5-31.

Drummond, M.E., Jr. "A Perspective on System Performance Evaluation." IBM Systems Journal, Vol. 8, No. 4, 1969, pp. 252-263.

Dunn, D.A. "Alternative Future Computer-Communication Markets." First International Conference on Computer Communications Proceedings, New York: Association for Computing Machinery, pp. 63-67.

Eddy, R.B. "The Role of Data Base Administration," Presented to Guide '34', May 17, 1972, (Eddy is Manager, Data Base Administration at Wellsco Data Corporation).

Emery, J.C. Organizational Planning and Control Systems: Theory and Technology. London: Crowell Collier and Macmillan, 1969.

Estrin, G., et. al. "SNUPER Computer: A computer in Instrumentation Automation." Spring Joint Computer Conference Proceedings, Vol. 30, 1967, pp. 645-656.

"Evaluate Your Computer Installation." Management Services, January/February, 1971, p. 37.

Fidlow, D. "System Trade-Offs: Dedicated and Dialed Networks." Datamation, April, 1973, pp. 51-54.

Flanagan, J.L., et. al. "Synthetic Voices for Computers." IEEE Spectrum, October, 1970, pp. 22-45.

Flores, I. Data Structure and Management. Englewood Cliffs, N.J.: Prentice-Hall, 1970.

Foster, C.C. "A View of Computer Architecture." Communications of the ACM, Vol. 15, No. 7, July, 1972, pp. 557-565.

Foster, C.C. Computer Architecture. New York: Van Nostrand Reinhold Co., 1970.

Frank, R.A. "Advantage of Real-Time Systems: Clean Up Errors as They Occur." Computerworld, February 21, 1973, p. 7.

Frank, R.A. "Hardware 'Available' for $60 Mini." Computerworld, April 11, 1973, p. 8.

Frank, R.A. "Multiple Minis Meaningful for Message Movement." Computerworld, December 13, 1972, p. 3.

Frank, R.A. "Phone Lines Prone to Compromise." Computerworld, December 6, 1972, p. 19.

Frank, R.A. "T/S Vendors Stress Security of Terminal, Net, CPU." Computerworld, December 6, 1972, p. 21.

Gammill, R.C. "Applications of Relational Data Structure Models in Man-Machine Systems." Proceedings of the ACM Conference, August, 1972, pp. 282-298.

Gaushell, D.J. "Error Control in Digital Transmission." Control Engineering, June, 1972, pp. 58-63.

Gaudion, E.J. "Today IBM Announces Yesterday." Datamation, February, 1973, pp. 53-54.

"Getting Out The Output: A CW Special Report." Computerworld, October 25, 1972, Supplement pages 1-8.

Goodman, A.F., ed. Computer Science and Statistics: Fourth Annual Symposium on the Interface--An Interpretative Summary. North Hollywood, Calif.: Western Periodicals, Co., 1971.

Grace, G.L. "Application of Empirical Methods to Computer-Based Systems." Journal of Applied Psychology. June, 1966, pp. 442-450.

Green, R.M., Jr., ed. Business Intelligence and Espionage. Homewood, Ill.: Dow Jones-Irwin, Inc., 1966.

"Growth in CRT Output Use Booming." Computerworld, September, 27, 1972, p. 4.

Gruenberger, F., ed. Critical Factors in Data Management. Englewood Cliffs, N.J.: Prentice-Hall, 1969.

Guide Share Data Base Requirements Group. Data Base Management System Requirements, November 11, 1970.

Haavind, R. "The Real-Time Wizard of Cars." Computer Decisions, February, 1973, pp. 8-18.

Hamilton, P. Espionage and Subversion in an Industrial Society. London: Hutchinson & Co., 1969.

Hammer, C. "Computer Communications: The Future." First International Conference on Computer Communications Proceedings, New York: Association for Computing Machinery, October 24-26, 1972, pp. 31-35.

Hammer, C. "Signature Simulation and Certain Cryptographic Codes." Communication of the ACM, Vol. 14, No. 1, January, 1971, pp. 3-14.

Hanlon, J. "Cybernetics Helps Chileans Keep Watch on Industry." Computerworld, February 28, 1973, p. 6.

Hanlon, J. "For Want of Cloth (and Backup) There Was Confusion at Airport." Computerworld, February 28, 1973, p. 8.

Harary, Frank. Graph Theory. Reading, Mass.: Addison-Wesley Publishing Co., 1969.

Harding, W.J. and Howell, T.H. "Performance Analysis of Error-Detecting-And-Correcting Codes in a Random-Error Environment." Proceedings of the ACM Annual Conference, August, 1972, pp. 265-277.

"Hardware Simulation Packages May Cut Use of Benchmarking." Computerworld, March 17, 1971, p. 19.

Hare, Van Court, Jr. Systems Analysis: A Diagnostic Approach. New York, Harcourt Brace Jovanovich, Inc., 1967.

Hare, Van Court, Jr., ed. "The New Data Base Task Group Report." Data Base, Vol. 3, No. 3, Special Issue, 1971.

Harper, W.L. "The Remote World of Digital Switching." Datamation, March 15, 1971, pp. 22-31.

Harrison, A. The Problem of Privacy in the Computer Age: An Annotated Bibliography, Vol. 1. Santa Monica: The Rand Corp., RM-5495-PR/RC, December, 1967.

Hart, L.E. "The Users Guide to Evaluation Products." Datamation, Vol. 16, No. 17, December 15, 1970, pp. 32-35.

Hart, L.E. and Lipovich, G.J. "Choosing a System Stethoscope." Computer Decisions, November, 1971, pp. 20-23.

Hatfield, D.J. and Gerald, J. "Program Restructuring for Virtual Memory." IBM Systems Journal, No. 3, 1971, pp. 168-191.

Head, R.V. "Automated System Analysis." Datamation, August 15, 1971, pp. 22-24.

Hersch, P. "Data Communications." IEEE Spectrum, February, 1971, pp. 47-60.

Hicks, H.T., Jr. "ANSI Cobol." Datamation, November 1, 1970, pp. 32-36.

"Hierarchical Minis Expected By 1975." Computerworld, April 25, 1973, p. 60.

"Hitachi Refines Laser Memory." Computerworld, October 18, 1972, p. 29.

Hitch, C. "On the Choice of Objectives in System Studies." In Eckman, D.P., ed., Systems: Research and Design. New York: Wiley, 1961.

Hitch, C. "Sub-optimization in Operations Problems." Journal of the Operations Research Society of America (now Operations Research), Vol. 1, No. 3, 1953, pp. 87-99.

Hoffman, L.J. "The Formulatory Model for Flexible Privacy and Access Control." Fall Joint Computer Conference Proceedings, 1971, pp. 587-601.

Holland, S.A. "The Remote Inquiry of Data Bases." Datamation, November 15, 1970, pp. 54-62.

Holt, A.W. Data Structure Theory and Techniques. Management Information Services, P.O. Box 5129, Detroit, Michigan 48236.

Holt, A.W. "Smart Terminals." Datamation, October 15, 1970, pp. 51-57.

Holt, R.C. "Some Deadlock Properties of Computer Systems." Computing Surveys, Vol. 4, No. 3, September, 1972, pp. 179-196.

Hootman, J.T. "The Computer Network as a Marketplace." Datamation, April, 1972, pp. 43-46.

Horning, J.J. and Randell, B. "Process Structuring." Computing Surveys, Vol. 5, No. 1, March, 1973, pp. 5-30.

"How Important the Data? How Much Speed?" Computerworld, April 11, 1973, p. 4.

Hughes, P.H. and Moe, G. "A Structural Approach to Computer Performance Analysis." National Computer Conference Proceedings, Vol. 42, 1973, pp. 109-120.

Huyck, P.H. "CAI Techniques for Information Retrieval." Datamation, February, 1973, pp. 91-92.

"IBM Bubbles May Be in Users Future." Computerworld, February 28, 1973, p. 18.

"IBM Continues to Lose Grasp on Federal Base; Univac No. 1." Computerworld, April 25, 1973, p. 59.

"IBM OCR Unit Aimed at Small Users." Computerworld, November 1, 1972, p. 15.

Inose, H. "Communication Networks." Scientific American, September, 1972, pp. 117-128.

"Inside Eastern's Data Center." Business Week, February 5, 1972, p. 60-61.

"Intermag-1971." Datamation, September 15, 1971, p. 37.

Introduction to Data Management: Student Text, IBM Corp., DPD Education Development, Poughkeepsie, N.Y. 12602, First Edition, April 1970.

Iverson, K.E. A Programming Language. New York: John Wiley & Sons, Inc., 1962.

Jones, C.H., Hughes, J.L. and Engvold, K.J. "A Comparative Study of Computer-Aided Decision Making From Display and Typewriter Terminals." Poughkeepsie, New York: IBM Systems Development Division Report, June 12, 1969.

Johnson, R.R. "Needed: A Measure for Measure." Datamation, Vol. 16, No. 17, December 15, 1970, pp. 22-30.

Johnson, R.R. "Some Steps Toward an Information System Performance Theory." Performance Evaluation Review, Vol. 1, No. 3, September, 1972, pp. 4-15.

Joslin, E.O. "Applications Benchmarks--The Key to Meaningful Computer Evaluations." Proceedings, ACM 20th National Conference, 1965, pp. 27-37.

Jurgen, R.K. "Minicomputer Applications in the Seventies." IEEE Spectrum, August, 1970, pp. 37-52.

Karush, A. "Performance Measurement." Data Management July, 1971, pp. 36-40.

Kernighan, B.W.; Plauger, P.J. and Plauger, D.J. "On Comparing Apples and Oranges, or My Machine is Better Than Your Machine." Performance Evaluation Review, Vol. 1, No. 3, September 1972, p. 16-20.

"Keypunch Replacement? 'But Let's Take It Easy.'" Computerworld, April 11, 1973, p. 4.

Kimbleton, S.R. "The Role of Computer System Models in Performance Evaluation." Communications of the ACM, July, 1972, pp. 586-590.

King, P.F. and Collmeyer, A.J. "Database Sharing--An Efficient Mechanism for Supporting Concurrent Processes." National Computer Conference Proceedings, Vol. 42, 1973, pp. 271-276.

Knuth, D.E. The Art of Computer Programming, Vol. 1. Reading, Mass.: Addison-Wesley Publishing Company, 1968.

Kobayashi, I. "An Algebraic Model of Information Structure and Information Processing." Proceedings of the ACM Annual Conference, August, 1972, pp. 1090-1104.

Kochem, M. and Deutsch, K.W. "Pluralization: A Mathematical Model." Operations Research, Vol. 20, No. 2, March-April, 1972, pp. 276-292.

Kolence, K.W. "Software Physics and Computer Performance Measurements." Proceedings of the ACM Annual Conference, 1972, pp. 1024-1040.

Kolence, K.W. "A Software View of Measurement Tools." Datamation, Vol. 17, No. 1, (January 1, 1971), pp. 32-38.

Kolence, K.W. "System Improvement by System Measurement." Data Base, Winter 1969, pp. 6-11.

Kretzmer, E.R. "Communication Terminals." Scientific American, September, 1972, pp. 131-140.

Kriebel, C.H. "MIS Technology--A View of the Future." Spring Joint Computer Conference Proceedings, 1972, pp. 1173-1180.

Kriebel, C.H. "Operations Research in the Design of Management Information Systems." Pittsburgh, Penn.: Graduate School of Industrial Administration, Carnegie Institute of Technology, Reprint #314, 1966.

Krinos, J.D. "Interaction Statistics from A Data Base Management System." National Computer Conference Proceedings, Vol. 42, 1973, pp. 283-289.

Kurtz, J.F. and Cuozzo, D.E. "How to Get Real Benefits from Virtual Storage." Datamation, pp. 48-52.

LaBlanc, R.E. and Himsworth, W.E. "Data Communications in 1890--A Capital Market View." Spring Joint Computer Conference Proceedings, 1972, pp. 611-616.

Lapidus, G. "A Glossary of IC Terms." Control Engineering, August, 1970, pp. 50-52.

Lapidus, G. "The Domain of Magnetic Bubbles." IEEE Spectrum, September, 1972, pp. 58-62.

Laska, R.M. "All the News That's Fit to Retrieve." Computer Decisions, August, 1972, pp. 18-22.

Lasser, D.J. "Productivity of Multiprogrammed Computers--Progress in Developing an Analytic Prediction Method." Communications of the ACM, Vol. 12, No. 12, December, 1969, pp. 678-684.

Leavitt, D. "DED Seen as Key to Data Management." Computerworld, March 21, 1973, p. 13.

Leavitt, D. "Encrypting Routines Offered, But Not Widely Used." Computerworld, December 6, 1972, p. 18.

Leavitt, D. "One Application May Justify Keypunch Replacement." Computerworld, February 21, 1973, p. 4.

Leavitt, D. "Performance Monitors Can Steal Data." Computerworld, April 18, 1973, p. 15.

Leavitt, D. "Probe Points--Whose Data?" Computerworld, p. 1 & 2.

Leavitt, D. "Programming Style Can Control Paging." Computerworld, November 28, 1972, p. 13.

Leavitt, D. "Retailers 'Married' to POS Terminals Appear Happy," Computerworld, December 6, 1972, p. 26.

Leavitt, D. "System Evaluation Must Be Planned." Computerworld, January 10, 1973, p. 8.

Leavitt, D. "Use Microprograms for Measurement." Computerworld, November 22, 1972, p. 11.

Leavitt, D. "Users Told to Evaluate Cassette, Language Features." Computerworld, February 21, 1973, p. 5.

Lefkovitz, D. File Structures for On-Line Systems. New York: Spartan, 1969.

Lee, J.A.N. The Anatomy of a Compiler. New York: Reinhold Book Corp., 1967.

Lucky, R.W. "Structure in Error-Correcting Codes." Hierarchical Structures. Edited by L.L. Whyte, A.G. Wilson and D. Wilson. New York: American Elsevier Publishing Co., Inc., 1969.

Lucas, H.C., Jr. "Performance Evaluation and the Management of Information Services." Data Base, Vol. 4, No. 1, Spring, 1972, pp. 1-8.

Lucas, H.C., Jr. "Performance Evaluation and Monitoring." Computing Surveys, Vol. 3, No. 3, September, 1971, pp. 79-91.

Lucas, H.C., Jr. "Synthetic Program Specifications for Performance Evaluation." Proceedings of the ACM National Conference, 1972, pp. 1041-1060.

Lundell, E.D., Jr. "'Inflexible' D.P. Systems Said to Attract Dishonesty." Computerworld, November 1, 1972, p. 2.

Lundell, E.D., Jr. "Magnetic Bubbles Place Seen Between Core and Disks." Computerworld, September 27, 1972, p. 24.

Lundell, E.D., Jr. "Measuring Tools Get GAO Blessing." Computerworld, September 6, 1972, p. 1 & 2.

Lundell, E.D., Jr. "Mixed-Media Growth Tied to Applications." Computerworld, April 4, 1973, p. 31.

Lundell, E.D., Jr. "More Secure (But Vulnerable) Systems Expected Soon." Computerworld, December 13, 1972, p. 13.

Lundell, E.D., Jr. "What System Parameters Should be Measured?" Computerworld, December 13, 1972, p. 9.

Lynch, W.C. "Do Disk Arms Move?" Performance Evaluation Review, December, 1972, Vol. 1, No. 4, pp. 3-16.

Lynch, W.C. "Operating System Performance." Communications of the ACM, July, 1972, p. 579-585.

Lyon, J.D. "The Role of the Data Base Administrator," Data Base, Vol. 3, No. 4, Winter, 1971, pp. 11-12.

MacDougall, M.H. "Computer System Simulation: An Introduction." Computing Surveys, Vol. 2, No. 3, September, 1970, pp. 191-210.

"Making the Case for Key-To-Disk." Computer Decisions, April, 1973, pp. 20-21.

"Managers Brace for Future DP Shock." Computerworld, April 4, 1973, p. 24.

Marschak, J. "Economics of Information Systems." Journal of the American Statistical Association, Vol. 66, No. 333, March, 1971, pp. 192-219.

Martin, J. Design of Real-Time Computer Systems. Englewood Cliffs, N.J.: Prentice-Hall, 1967.

Martin, J. Telecommunications and the Computer. Englewood Cliffs, N.J.: Prentice-Hall, 1969.

Martin, J. Teleprocessing Network Organization. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1970.

Martin, L.D. "A Model for File Structure Determination for Large On-Line Data Files." International Federation of Information Processing Society, Occasional Publication No. 3, File Organization, Selected Papers from File 1968--An I.A.G. Conference, Swets and Zeitlinger N.V. Amsterdam, 1969, pp. 223-245.

Mason, R.O. and Mitroff, I.I. "A Program for Research on Management Information Systems." Presented at the ORSA-TIMS-AIIE Joint National Meeting, Atlantic City, New Jersey, November 8-10, 1972.

Matthews, D.Q. The Design of the Management Information System. New York: Auerbach, 1971.

McFadden, T.F. and Strauss, J.C. "Quantitative Evaluation of File Management Performance Improvements." National Computer Conference Proceedings, Vol. 42, 1973, pp. 63-68.

McLaughlin, R.A. "IBM's Virtual Memory 370's." Datamation, September, 1972, pp. 58-60.

Mealy, G.H. "Another Look At Data." Fall Joint Computer Conference Proceedings, 1967, pp. 525-534.

Meiners, E.E. "A Machine-Independent Data Management System." Datamation, June 1973, pp. 92-95.

Mellen, G.E. "Cryptology, Computers and Common Sense." National Computer Conference Proceedings, Vol. 42, 1973, pp. 569-580.

Menck, H.R., editor. "SIGMETRICS Bibliography." New York: Association for Computing Machinery, August 1, 1971.

Menkus, B. "Retention of Data for the Long Term." Datamation, September 15, 1972, pp. 30-32.

Merritt, M. "Human Factors Need Work." Computerworld, February 14, 1973, p. 4.

Mesniaeff, P.G. "The Technical Ins and Outs of Computerized Numerical Control." Control Engineering, March, 1971, pp. 65-84.

Meyer, C.H. "Design Considerations for Cryptography." National Computer Conference Proceedings, Vol. 42, 1973, pp. 603-606.

Miller, E.F., Jr. "Bibliography on Techniques of Computer Performance Analysis." Computer, September/October, 1972, pp. 39-47.

Morris, R. "Scatter Storage Techniques." Communications of the ACM, Vol. 11, No. 1, January, 1968, pp. 38-44.

Morton, M.S.S. Management Decision Systems: Computer-Based Support for Decision Making, Boston: Division of Research, Graduate School of Business Administration, Harvard University, 1971.

Moulder, R. "An Implementation of a Data Base Management System on an Associative Processor." National Computer Conference Proceedings, Vol. 42, 1973, pp. 171-176.

National Bureau of Standards. Guidelines for Describing Information Interchange Formats. Washington, D.C.: Superintendent of Documents, U.S. Government Printing Office, SD Catalog No. C 13.52:20, March 1, 1972.

Naylor, T.H., ed. The Design of Computer Simulation Experiments. Durham, N.C.: Duke University Press, 1969.

Nemeth, A.G. and Rovner, P.D. "User Program Measurement in a Time-Shared Environment." Communications of the ACM, Vol. 14, No. 10, October, 1971, pp. 661-666.

Nolan, R.L. "Managing the Data Resource: The Organization Issue." CBIS Working Paper #2, Boston, Mass.: Graduate School of Business Administration, Harvard University, Revision #3, May 7, 1973.

Nolan, R.L. "Systems Analysis for Computer-Based Information Systems Design." Data Base, Vol. 3, No. 4, Winter, 1971, pp. 1-10.

Norman, J.L. "Reducing Telephone Network Errors." Datamation, October 1, 1971, pp. 24-31.

"OCR Devices Mean 'Gravy'" Computerworld, February 21, 1973, p. 4 & 5.

"OCR Market Expected to Grow, Potential Of Key-Entry Combination Also Cited." Computerworld, November 29, 1972, p. 52.

Olle, T.W. An Assessment of How the CODASYL Data Base Task Group Proposal Meets the Guide-Share Requirements. Publication Department, Norwegian Computing Center, Forksningsveien 1B, Oslo 3-Norway.

Olle, T.W. "MIS: Data Bases." Datamation, November 15, 1970, pp. 47-50.

Oppenheimer, G. and Weizer, N. "Resource Management for a Medium Scale Time-Sharing Operating System." Communications of the ACM, Vol. 11, No. 5, May, 1968, pp. 1-10.

"Optical Systems Eye-Openers." Computerworld, January 17, 1973, p. 3.

Ore, O. Graphs and Their Uses. New York: Random House, 1963.

Ovanski, D. "Modern Tools of Analysis." Data Management, May, 1971, p. 30-31.

Peck, P.L. "Effective Corporate Networking, Organization and Standardization." Fall Joint Computer Conference Proceedings, 1971, pp. 561-569.

Pendleton, J.C. "Integrated Information System." Fall Joint Computer Conference Proceedings, 1971, pp. 491-500.

Petroff, J.N. "Why are DP Managers So Unpopular?" Datamation, February, 1973, pp. 77-79.

Plagman, B.K. and Altshuler, G.P. "A Data Dictionary/Directory System Within the Context of an Integrated Corporate Data Base." Fall Joint Computer Conference Proceedings, 1972, pp. 1133-1140.

Proceedings of the 1972, Carnahan Conference on Electronic Crime Countermeasures. University of Kentucky: College of Engineering, April, 1972.

Propokop, J.S. and Brooks, F.P., Jr. "Decision Making With Computer Graphics in an Inventory Control Environment." Fall Joint Computer Conference Proceedings, 1970, pp. 599-607.

Pryke, J.T.M. "A Front-End Primer for IBM Users." Datamation, April, 1973, pp. 46-50.

Punchard, J.C.R. "What's Ahead in Communications?" IEEE Spectrum, January, 1970, pp. 51-54.

Reagan, F.H., Jr. "Is the Keypunch Really Obsolete?" Computer Decisions, April, 1973, pp. 12-17.

Renouard, C.A. "A Manufacturing Information and Control System Design." Control Engineering, September, 1970, pp. 78-81.

"RCA Uses Holographic Approach, IBM Puts 8K on Chip." Computerworld, February 28, 1973, p. 43.

Rogers, W.P. "Instant Financial Information." Datamation, February, 1973, pp. 65-68.

Rudin, H., Jr. "Data Transmission: A Direction for Future Development." IEEE Spectrum, February, 1970, pp. 74-85.

Ruth, S.S. and Kreutzer, P.J. "Data Compression for Large Business Files." Datamation, September, 1972, pp. 62-66.

Ruud, R.J. "The CPM-X--A Systems Approach to Performance Measurement." Fall Joint Computer Conference, Vol. 41, 1972, pp. 949-957.

"Sabotage Was the Problem." Computerworld, August 2, 1972, p. 1.

Sackman, H. Man-Computer Problem Solving. New York: Auerbach Publishers, Inc., 1970.

Sahin, K.E. An Investigation into Non-Directed Thought Processes and Their Pertinence to Information Systems Processing Intelligence--A Working Paper. Boston: Sloan School of Management, M.I.T., May 17, 1968.

Salasin, J. "Hierarchical Storage in Information Retrieval." Communications of the ACM, Vol. 16, No. 5, May, 1973, pp. 291-295.

Saltzer, J.A. and Gintell, J.W. "The Instrumentation of MULTICS." Communications of the ACM, Vol. 13, No. 8, August, 1970, pp. 495-500.

Schorr, H. and Waite, W.M. "An Efficient Machine-Independent Procedure for Garbage Collection in Various List Structures." Communications of the ACM, Vol. 10, No. 8, August, 1967, pp. 501-506.

Schubert, R.F. "Basic Concepts in Data Base Management Systems." Datamation, July, 1972, pp. 42-50.

Schulman, F.D. "Hardware Measurement Device for IBM System/360 Time Sharing Evaluation." Proceedings, ACM National Meeting, 1967, pp. 103-109.

Schumacher, D. "First Annual SIGME Symposium." Datamation, May, 1973, pp. 91-93.

Schwartz, J.M. and Wyner, D.S. "Use of the SPASM Software Monitor to Evaluate the Performance of the Burroughs B6700." National Computer Conference Proceedings, Vol. 42, 1973, pp. 93-100.

"Security Device Market Still in Infancy." Computerworld, December 6, 1972, p. 47.

Sedgewick, R.; Stone, R; and McDonald, J.W. "SPY: A Program to Monitor OS/360." Fall Joint Computer Conference, Vol. 37, 1970, pp. 120-128.

Senko, M.E., et. al. "Data Structures and Accessing in Data-Base Systems." IBM Systems Journal, Vol. 12, No. 1, 1973, pp. 30-93.

Seven, M.J.; Boehm, B.W.; and Watson, R.A. A Study of User Behavior in Problem Solving With An Interactive Computer. Santa Monica: Rand, R-513-NASA, April, 1971.

Severance, D.G. and Merten, A.G. "Performance Evaluation of File Organizations Through Modeling." Proceedings of the ACM National Conference, 1972, pp. 1061-1072.

Sewald, M.D.; Rauch, M.E.; Rodiek, L. and Wertz, L. "A Pragmatic Approach to Systems Measurement." Computer Decisions, July, 1971, pp. 38-40.

Shannon, C.E. and Warren, W. The Mathematical Theory of Communication. Urbana, Ill.: The University of Illinois Press, 1949.

Shneiderman, B. "Optimum Data Base Reorganization Points." Communications of the ACM, Vol. 16, No. 6, June, 1973, pp. 362-365.

Simon, H.A. Administrative Behavior. New York: The Free Press. London: Collier-MacMillian Ltd., 1945.

Simon, H.A. The Sciences of the Artificial. Cambridge, Mass.: MIT Press, 1969.

Smalheiser, M. "Airline Switches and Saves $350,000/Yr." Computerworld, April 18, 1973, p. 8.

Smith, J.L., et. al. "An Experimental Application of Cryptography to a Remotely Accessed Data System." Proceedings of the ACM Annual Conference, August, 1972, pp. 282-298.

Smith, W.H. "PCM Microwave Links." Telecommunications, April, 1973, pp. 25-32.

Sollenberger, H.M. Major Changes Caused by the Implementation of a Management Information System. New York: National Association of Accountants, 1968.

Stefferud, E. "Management's Role in Networking." Datamation, April, 1972, pp. 40-42.

Stogdill, Ralph M., ed. The Process of Model Building in the Behavioral Sciences. Columbus, Ohio: Ohio University Press, 1970.

Stutzman, B.W. "Data Communication Control Procedures." Computing Surveys, Vol. 4, No. 4, December, 1972, pp. 197-220.

"System Protection Depends on Well-Educated Auditor." Computerworld, March 28, 1973, p. 8.

Teichroew, D. and Sayani, H. "Automation of System Building." Datamation, August 15, 1971, pp. 25-30.

Tien, C.C. "Distributed Intelligence for Use-Oriented Computing." Fall Joint Computer Conference Proceedings, 1972, pp. 1049-1056.

"The Mighty Minicomputer: A CW Special Report." Computerworld, August 30, 1972, Supplement Pages 1-15.

"The Poor Man's Magnetic Bubble Memory." Business Week, April 29, 1972, p. 80.

"The Retailers Go Electronic." Business Week, August 14, 1972, pp. 38-43.

"Traffic in Stolen Credit Cards May Stop with Terminal System." Computerworld, April 25, 1973, p. 49.

Tuffe, O.N. and Chen, D. "Optical Memories: Controlling the Beam." IEEE Spectrum, March, 1973, pp. 48-53.

Tufte, O.N. and Chen, D. "Optical Techniques for Data Storage," IEEE Spectrum, February, 1973, pp. 26-32.

Turoff, M. "Delphi and Its Potential Impact on Information Systems." Fall Joint Computer Conference Proceedings, 1971, pp. 317-326.

"TWA System Allows Continuous Processing." Computerworld, April 18, 1973, p. 4.

"Unit Puts Data Entry at Fingertips." Computerworld, August 16, 1972, p. 22.

Upton, M. "With On-Line Data Entry Users Work to Keep System Up." Computerworld, April 18, 1973, p. 4.

"Users Ride 'Intelligent' Bandwagon." Computerworld, April 25, 1973, p. 11.

Van Slyke, R.M.; Chow, W.; and Frank, H. "Avoiding Simulation in Simulating Computer Communications Networks." National Computer Conference Proceedings, Vol. 42, 1973, pp. 165-170.

Vazsonyi, A. "Why Should the Management Scientist Grapple with Information Systems." Presented at the Joint National Meeting of the Institute of Management Sciences, Operations Research Society of America, American Institute of Industrial Engineers, Atlantic City, New Jersey, November 8-10, 1972.

Ware, W.H. "The Ultimate Computer." IEEE Spectrum, March, 1972, pp. 84-91.

Warner, C.D. "Monitoring: A Key to Cost Efficiency." Datamation, Vol. 17, No. 1, January 1, 1971, pp. 40-42, 49.

Warner, C.D. "System Performance and Evaluation--Past, Present and Future." Fall Joint Computer Conference, Vol. 41, 1972, pp. 959-964.

Watson, R.A. Computer Performance Analysis: Applications of Accounting Data. Santa Monica: Rand Corp., R-573-NASA/PR, May, 1971.

Weinstein, M. "Key-Disk Best for Central Uses." Computerworld, April 4, 1973, p. 4.

Weinstein, M. "Key-Disk? Discard Safe Approach." Computerworld, April 11, 1973, p. 4.

Weinstein, M. "Minicomputer Trade-Offs are Changing." Computerworld, April 11, 1973, p. 21.

Weinstein, M. "Power Fluctuations can Cause Users More EDP Problems Than Brownouts." Computerworld, November 8, 1972, p. 16.

Weinstein, M. "Selecting Communications Lines? Flowchart Needs." Computerworld, March 14, 1973, p. 6.

Weinstein, M. "Who Accesses What on Remote Terminal?" Computerworld, December 6, 1972, p. 22.

Weizer, N. and Oppenheimer, G. "Virtual Memory Management in a Paging Environment." Spring Joint Computer Conference Proceedings, Vol. 34, 1969, pp. 11-17.

"What's in the World of Input: A CW Special Report." Computerworld, September 27, 1972, Supplement Pages 1-19.

White, L.J. "Optimum Concentrator Location in Telecommunications Design." Proceedings of the ACM Annual Conference, August, 1972, pp. 534-542.

Whitney, V.K.M. "Fourth Generation Data Management Systems." National Computer Conference Proceedings, Vol. 42, 1973, pp. 239-244.

"Why IBM Got the Jump in Bubble Memories." Business Week, February 17, 1973, pp. 38-40.

Weiner, N. Cybernetics. Paris: Hermann, 1958.

Wilkes, M.V. "The Growth of Interest in Microprogramming: A Literature Survey." Computing Surveys, Vol. 1, No. 3, September, 1969.

Williams, J.G. "Asymmetric Memory Hierarchies." Communications of the ACM, April, 1973, Vol. 16, No. 4, p. 213-222.

Williams, R. "A Survey of Data Structures for Computer Graphics Systems." Computing Surveys, Vol. 3, No. 1, March, 1971, pp. 1-21.

"Worldwide Spy Net on the Horizon." Computerworld, March 7, 1973, p. 4.

"Would-Be Net Users Hurt By Poor Cost Planning." Computerworld, March 21, 1973, p. 4.

Zannetos, Z.S. "On The Theory of Divisional Structures: Some Aspects of Centralization and Decentralization of Control and Decision Making." Management Science, Vol. 12, No. 2, December, 1965, pp. B49 - B68.

Zeigler, W.T. "OCR Relieves Small User's Conversion Problems." Computerworld, December 6, 1972, p. 38.