DOCUMENT RESUME

ED 236 974 HE 016 659

| | |
|---|---|
| AUTHOR | Chachra, Vinod; Heterick, Robert C. |
| TITLE | Computing in Higher Education: A Planning Perspective for Administrators. CAUSE Monograph Series. |
| INSTITUTION | CAUSE, Boulder, Colo. |
| PUB DATE | 82 |
| NOTE | 201p. |
| AVAILABLE FROM | CAUSE Publications, 737 Twenty-Ninth Street, Boulder, CO 80303 ($6.00, members; $10, nonmember colleges; $20, others). |
| PUB TYPE | Guides - Non-Classroom Use (055) -- Viewpoints (120) |
| EDRS PRICE | MF01 Plus Postage. PC Not Available from EDRS. |
| DESCRIPTORS | Administrative Principles; Administrator Guides; *College Administration; College Planning; *Computer Oriented Programs; Computer Programs; Databases; *Data Processing; Economic Climate; Higher Education *Information Networks; *Management Information Systems; Microcomputers; Minicomputers; Program Administration; Technological Advancement; Word Processing |

ABSTRACT

Options for using computers in managing higher education institutions and technological questions are considered in a collection of nine essays developed by the authors for this monograph. An introduction considers historical developments and provides an overview of computing modes and languages. After considering some of the economic and technological influences under which computing centers of the 1980s will have to operate, attention is directed to the following concerns: the question of whether to do different tasks on the same machine, management techniques, the reporting structure of the information systems department, job shop scheduling versus utility operation, development versus maintenance, and evaluation and audits. Topics concerning software development include models and design of data-based management systems, systems development life cycle, and structured techniques. Operations concerns are also covered, including scheduling, critical paths, service charging, job accounting, and queuing and simulation. The role of the user and the system view are considered, along with perspectives on office systems and word processing, and planning for the automated office. Issues concerning networks and sharing are also discussed, including the impact of the micro/mini computers and software sharing. Perspectives on the future of administrative computing are also considered. (SW)

CAUSE

MONOGRAPH
SERIES

# Computing In Higher Education: A Planning Perspective For Administrators

By
Vinod Chachra
and
Robert C. Heterick



The Professional Association
for Development, Use, and
Management of Information Systems
in Higher Education

**Library of Congress Cataloging in Publication Data**

Chachra, Vinod.
Computing in higher education.

(CAUSE monograph series)
1. Education, Higher--Data processing--Planning--Handbooks, manuals, etc. I. Heterick, Robert C., 1936- . II. Title.
LB2324.C48 1982 378'.0028'5 82-22989

Copies of this monograph are available to staff of CAUSE member institutions at $6 per copy, to non-member colleges and universities at $10, and to others at $20. Orders should be pre-paid and addressed to:

CAUSE Publications
737 Twenty-Ninth Street
Boulder, Colorado 80303
(303) 449-4430



Printed in the United States of America.

Library of Congress Catalog number 82-22989.

# *Preface*

A word of explanation is in order for the reader about to embark upon pursuing the contents herein. The reader will discover a loose collection of essays dealing with what is commonly called "Administrative Data Processing." The collection has resulted from the oft expressed desire to bring together the management of the College or University to discuss the ways, means and idosyncracies of computer-aided institutional management.

When the opportunity presented itself to conduct a series of professional development seminars dealing with the computer and administration in higher education, we drew upon a stock of previous articles and position papers to construct the nucleus of this present work. We both are products of the computer revolution. We have studied, taught and managed educational computing for the entirety of our professional careers. Our viewpoints have been shaped not only by our management experiences (primarily in larger institutional settings) but also by our classroom challenges.

We have not attempted to produce a "how to" manual, but rather to cast a host of technological questions in a management perspective. Neither of us feels there is a "right way" to conduct administrative data processing, only that there is "a way" that best compliments and enhances the distinctive management style of a given institution at a particular time in its development. Because institutional administration is dynamic, it is useful, perhaps necessary, to be aware of the range of hardware, development, organization, etc. options for administrative computing.

We wish to acknowledge the financial assistance of the IBM Corporation in providing the support to CAUSE for the initial distribution of this work. We need also to acknowledge the careful criticism of the style of this work by Julia A. Rudy of CAUSE and the thoughtful construction of the figures by Fred W. Lacerda. The text and figures were produced by the authors using APL and SCRIPT as part of a test documentation system currently under development. Any errors in the finished work are solely of our own construction.

V. Chachra
R. C. Heterick

Blacksburg, Virginia
January 1982

# About CAUSE

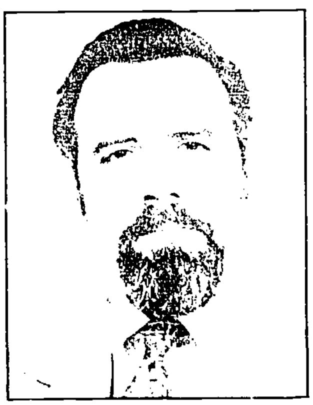
CAUSE, the Professional Association for Development, Use, and Management of Information Systems in Higher Education, helps member institutions strengthen their management capabilities through improved information systems.

CAUSE provides member institutions with many services to increase the effectiveness of their administrative information systems. These services include: the Exchange Library, which is a clearinghouse for non-proprietary information and systems contributed by members; an Information Request Service to locate specific systems or information; consulting services to review AIS organization and management plans; organizational publications including a bi-monthly newsletter, a bi-monthly professional magazine and the CAUSE Monographs; and the CAUSE National Conference.

Additional details on the CAUSE organization and activities may be obtained from the CAUSE National Office in Boulder, Colorado.

# About the Authors

Vinod Chachra is the Assistant Provost for Planning and Information Systems at Virginia Tech. In this position, he is responsible for academic planning as well as direct supervision of the University Library, the University Computing Center and the University Registrar. In addition he serves as the director for the Center for Library Automation. He has several professional publications to his credit and is co-author of a book on graph theory published by American Elsevier. He was elected as delegate to the White House Conference on Library and Information Sciences. Dr. Chachra is a past president of CAUSE.



Robert C. Heterick, Jr. is Director of the Design Automation Laboratory at Virginia Tech. Prior to his current position, Dr. Heterick was Director of the Computing Center at the University. He has lectured widely on computer systems management and is the author of several papers on the subjects of resource allocation in computing systems and decision support systems. He has served as a consultant to the IBM Corporation, the Governor of Virginia, the President's Privacy Protection Study Commission and numerous universities and private corporations on computer systems management and distributed computing in networks of microprocessors.

# Table of Contents

# 1

# COMPUTING PERSPECTIVES

"Give me a lever long enough . . .
and I can move the world singlehanded."

Archimedes

## Archimedes Lever

There now exists a whole generation of adults who have lived their lives with computer technology. It is sometimes difficult to view computing in its proper perspective, at the same time taking into account its old recorded history and today's new headlines. Society is just now beginning to come to grips with the impact of the digital computer on our life styles. As with any new technology it has come from the phase of derided laboratory apparatus, through oversold savior of mankind to potential oppressor of human dignity. Like most new technologies it is viewed somewhat ambivalently, indicating we still have far to go before its total acceptance.

Within the short span of 25 years, the digital computer has gone through four technological generations – from large, expensive, hot, vacuum tube devices to portable, inexpensive (less than the cost of a standard automobile), virtual devices capable of being addressed in near natural language. Ambivalence toward computers is recognized when we reflect that without them we would not have put a man on the moon, we could not operate our current financial institutions, near instantaneous airline reservations would be impossible and many of our newer structures could not be built or operated. At the same time they have brought us seemingly irreconcilable billing foul-ups, systems which do not work when the power fails and potential for privacy invasion on a heretofore undreamed of scale. The positives can be attributed to technology, the negatives to man's ability to understand and control that technology. Many of our perceived problems with computers probably stem from our attempt to use the new technology in old ways, much as we installed buggy whips on automobiles at the turn of the century.

The potential of the computer has been well expressed in terms of multipliers by W. J. McKeefery.[1]

> For centuries, man worked to make multipliers improve his own strength and skill by factors of three, five or ten. The hoe, the wheel, the bow and arrow represented modest gains. Then, with the industrial revolution, the multipliers became a hundred to a thousand-fold. Gunpowder, steam power, fossil fuels pushed the horsepower per person into this range and stretched his travel comprehension in approximately these measures. Even today, jet transportation does not improve walking by more than a thousand.
>
> Beyond all this, there comes along occasionally a multiplier in the range of a million to one. These have been recent and still relatively rare. The first multiplier came earlier in this century by controlling electricity. First in wires, and then by wireless, the speed of transmission gained is in the order of a million to one. Later, through the multiplier of mass production of radio and television sets, these swift moving messages were placed in millions of homes simultaneously.
>
> Another major multiplier in the million category is nuclear energy, which began with shattering explosions in the form of bombs, but now, hopefully, will be developed as controlled fusion for the major

---

[1] W.J. McKeefery, "Where are We? A Top Management Perspective," in *Organizing for Innovation: Towards the Computer Utility*, ed. R. C. Heterick (Blacksburg: Virginia Tech Computing Center, 1973).

energy source of the world's population. Today, nuclear energy in power plants and marine propulsion has reached the thousand-fold category. With fusion it will reach a million fold and gradually change our way of life.

Still another important multiplier is the computer. It, too, has reached the million fold category. Its combination of speed of calculation in nano-seconds and its storage capacity make its superiority over man in the million fold category. We haven't exploited this leverage fully yet, but with miniaturization, portability and the development of symbolic languages along with networks, these combined resources will produce again one of the vast changes in mankind's way of doing business.

When we consider the potential of the computer in communication networks we begin to achieve a million multiplication of a million multiplier. The potential of such an environment will probably take a generation to make its true impact felt. When we pause to consider that commercial aviation represents only two orders of magnitude improvement over walking, and reflect on the tremendous impact it has had on our life styles and comprehension of the world we live in, we cannot help but be awed in trying to comprehend the impact of the computer. In just the last 20 years the computer has demonstrated four to six orders of magnitude improvement in computational speed, printing speeds, packing densities and the cost to process and transmit information.

## A Brief Look Backward

Man has been computing for the thousands of years of recorded history. He started it all by learning to count. Beginning with his fingers and toes as counters, he probably next used marks, pebbles and progressed to beads. As his needs became more complex he developed symbolic notation for representing the things he had counted. Much later, he developed mathematical systems for manipulating his symbols. With one notable exception, the abacus (a counting frame developed about 3000 B.C.), man made very little progress toward the development of mechanical counters until about 1640, when Pascal developed a gear-driven calculator. It was not until 1920 that the desk calculator was given an electric drive.

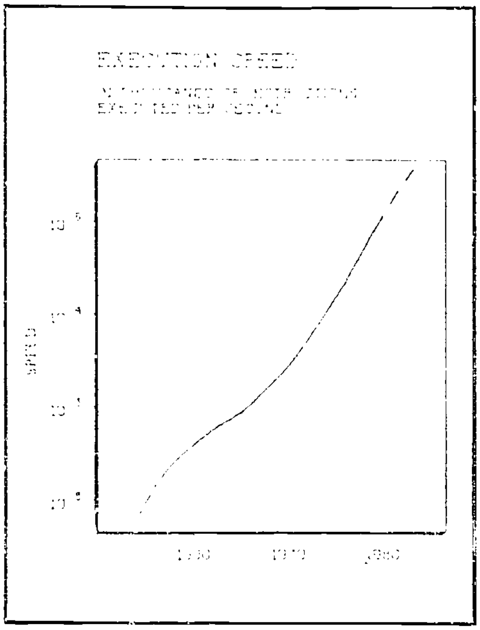



Figure 1
The improvement of
execution speeds

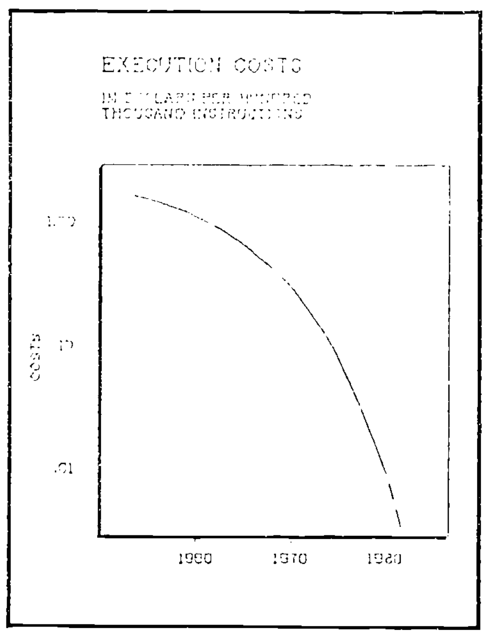



Figure 2
The improvement in
execution costs

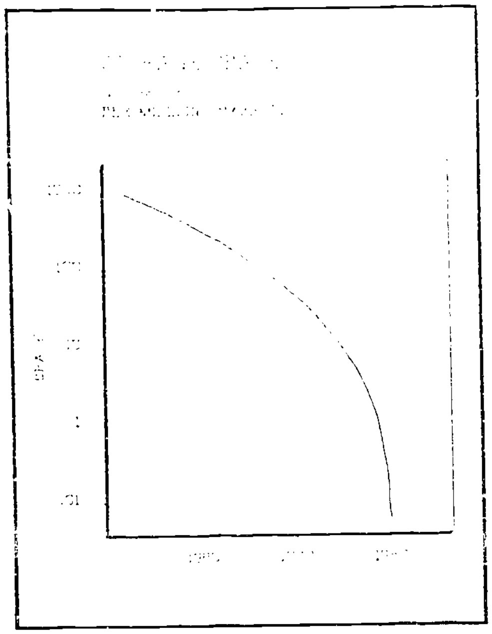


Figure 3
Size effect of
technology on
storage space

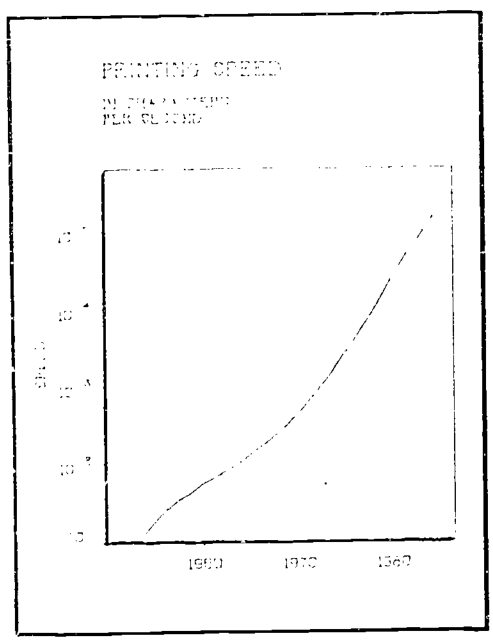



Figure 4
Increases in
printing speed

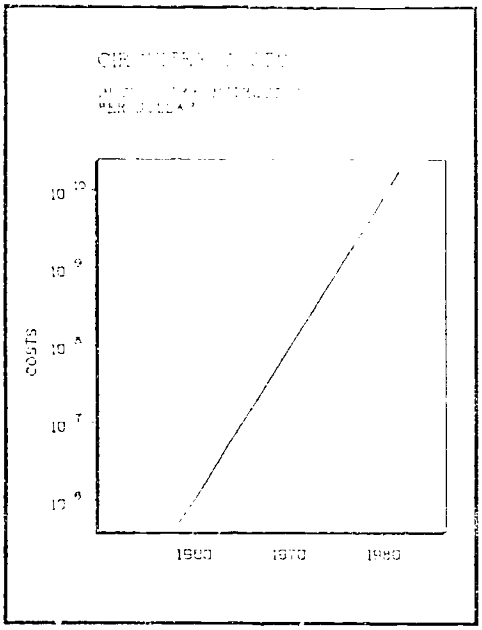



Figure 5
Decrease of
circuit costs

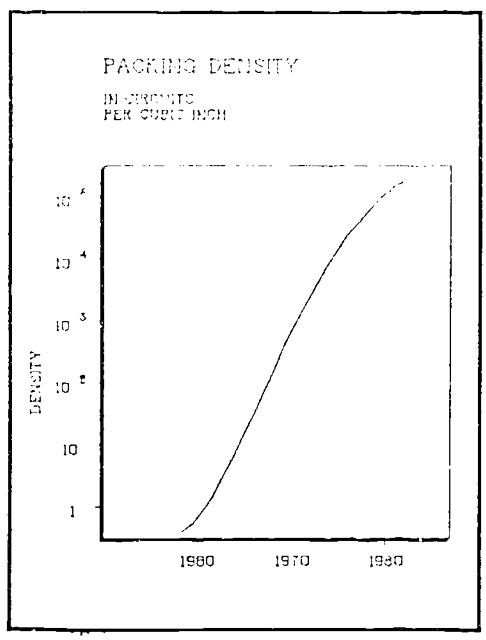



Figure 6
Increase of
packing density

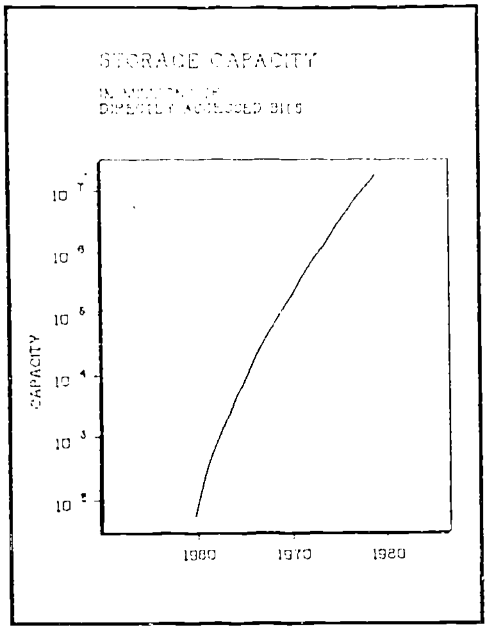



Figure 7
Increase in directly accessed storage capacity

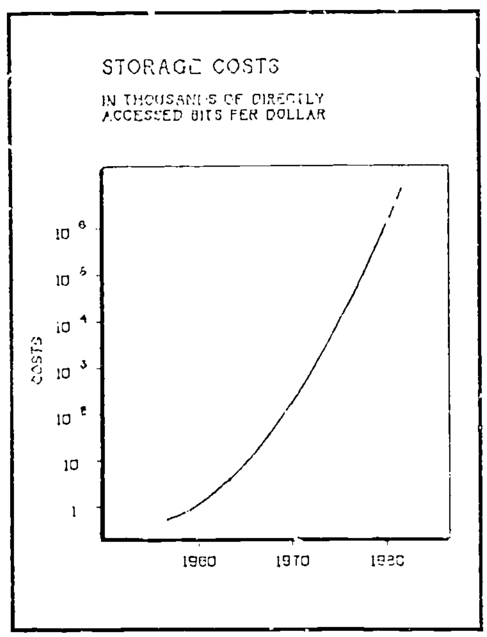



Figure 8
Decrease in directly accessed storage costs

In the early 1800s, Jacquard developed an "automatic" loom controlled by punched cards. The Jacquard loom was the predecessor of modern tabulating equipment pioneered by Herman Hollerith of the U. S. Census Bureau. In 1880, recognizing that it would take over ten years to process the decennial census data, Hollerith began working on the idea of electro-mechanical counters. The standard tabulating card was first cut by the Bureau of Engraving from machines made for cutting the old greenback dollar, hence its unusual size. These electro-mechanical devices, sorters, collators, tabulators, card punches, etc. were developed and marketed by the IBM Corporation and comprised the major form of business automation through the first half of this century.

During the early half of the 19th century, Charles Babbage worked on the forerunner of our modern electronic computer. With support from the British government and later from Lord Byron's daughter, Lady Lovelace, he developed two computers, the difference engine and the analytical engine. Neither computer worked, primarily because the rather sophisticated machine parts could not be manufactured precisely enough. The analytical engine was built to Babbage's specification over 100 years later and worked as he envisioned it would.

As problems became more numerous and more complex, World War II provided a definite impetus, man applied more and more effort toward the development of machine and electrical and electronic devices which would assist him in his problem solving efforts. Those devices which calculate by counting are called digital devices.

Man learned another way to compute, also. He discovered that he could solve some of his problems by measuring. His first efforts at such computation were probably direct, i.e., the problem was solved by actual measurement of the quantity being studied. Later he learned to represent a given system by another which was similar (or analogous) in one or more respects to the original (or prototype) system. These devices he called analogs. Analog devices include the slide rule, scale models, computing pumps and calculating scales. Some of these devices are direct analogs in that quantities being measured are of the same type or have the same units as quantities in the system being studied. Some are indirect analogs with measured quantities differing greatly from the desired quantities.

Around 1600, Galileo developed a set of scales (an analog device) for aiming artillery pieces. The scales served to solve the second order equation of the motion of a projectile. About the same time, Napier developed logarithms. Scales ruled according to the logarithms of numbers were soon used as analog computers and the slide rule gained the prominence it was to hold until the 1960s. The cost of a hand-held electronic calculator had dropped to under $10 by 1975. These calculators were capable of nearly all the operations of a conventional slide rule at one-third the price. The last major supplier of slide rules ceased production of that portion of their product line in 1976.

In 1876, Lord Kelvin and his brother, James Thompson, developed an analog computer for predicting tides. The tidal analyser required mechanical apparatus to compute areas (the process of integration) and was further refined to solve ordinary differential equations. Later versions of the analyzer were capable of harmonic analysis.

In 1936, the brilliant English mathematician, A.M. Turing, presented a landmark paper in which he described a machine which could simulate any other automaton. The Turing machine consisted only of the concept of an infinitely long tape, three symbols and a place marker. He demonstrated that this machine could simulate the behavior of any other machine, no matter how large its symbol or instruction repertoire. With this paper was born the field of artificial intelligence and the concept of "thinking machines."

One other development of the mid-1940s was particularly significant in the evolution of computers. The late Dr. John von Neumann, a mathematician who served both the Atomic Energy Commission and the Army Ordinance Department as a consultant, was aware of the AEC's computational needs and the Army's interest in the ENIAC. He proposed the idea of a digital computer which could store both instructions and data internally in numeric form and perform numerical operations upon both types of stored information. Modification of instructions, he reasoned, would make it possible to achieve conditional solutions; i.e., solutions in which the path through the predefined instruction set could be dependent upon some previously computed result. Dr. von Neumann's idea sparked several new computer designs and the first computer utilizing this internal storage of both data and instructions became operational in 1949. The machine which von Neumann proposed was not completed until 1952, however. Today all general purpose digital computers have this capability.

The term "cybernetics" was coined by the eminent mathematician Norbert Wiener from the Greek word for steersman. Weiner put forth the bold proposition[2]

> ...that society can only be understood through a study of the messages and the communications facilities which belong to it; and that in the future development of these messages and communication facilities, messages between man and machines, between machines and man, and between machine and machine, are destined to play an ever-increasing part.

The broad area of man-machine feedback mechanisms is subsumed in cybernetics, as well as the more scientific disciplines of control and communications theory. Weiner's prophecy was recognized in less than one generation from the time he made it. We think nothing of computers as a constituent part of the family automobile, regulating fuel consumption, exhaust emission, etc. Few, if any, major systems are designed which do not feature some aspect of the man machine feedback mechanism.

## Analog Applications

The general purpose analog computer industry has grown rapidly but not nearly so rapidly as the digital industry. There are several reasons for the different growth rates. For one, the results obtained with the analog computer are not as precise as those from the digital. This is not a matter of concern to engineers or scientists because, generally speaking, the data in problems in their areas are not exact. Second, even if the results of analog computation were accurate enough for business operations (bookkeeping, inventory control, etc.), the analog computer does not have the data handling facilities for performing counting type operations. Third, the use of the analog computer requires a higher level of mathematical sophistication and competence at the introductory level than does the digital computer.

The analog computer is an extremely valuable tool for solving certain types of problems, particularly those which are peculiar to engineering. With it, systems which might cost large sums of money to build and test can be simulated at relatively low cost. Systems which might endanger life

---------------

[2] Norbert Weiner, *The Human Use of Human Beings*, (New York: Avon Books, 1950), p. 25.

or cause destruction of property if a test proved a design to be faulty (consider, for example, the consequence of failure of an experimental nuclear reactor) can be simulated on an analog computer in perfect safety. The analog computer can be programmed to run in a scaled time mode which either compresses or extends solution time. With such a program a space probe of many months duration may be simulated in minutes or an explosive type of action may be effectively slowed down to permit detailed study.

Computer users have generally belonged to one camp or the other with respect to their computational needs. They recognized fairly early that some types of problems yielded more easily to the analog and others to the digital approach. More recently, engineers in particular have found that the best way of solving some of their problems is to use both types of computers. The assemblage of the two types of comp'ters and the necessary linkage is called a hybrid comput.r.

Sensor based computer applications are becoming increasingly commonplace. Many of these applications utilize computing features that might most appropriately be called hybrid. At a minimum they feature analog to digital and digital to analog capabilities to sense and control environmental parameters, machine tool speeds, etc. Many modern industrial processes would be impossible without sensor-based, computer controlled, feedback mechanisms. It is probably in the area of hybrid applications that the full potential of the computer will be recognized. Administration in institutions of higher education will be extensively impacted by the application of analog devices in the campus computing environment. In addition to the obvious application of wand reading in the library and bookstore, we can expect more extensive use of the student identification card in machine read processes. Graphics, voice recognition and other analog applications will become the norm during the next ten years.

## The Modern Computer

Work on two unique electro-mechanical devices began prior to World War II and they became operational in 1942 and 1943, respectively. The first, Dr. Vannevar Bush's Differential Analyzer (Model II) was built at MIT and was the first successful large scale analog computer capable of sufficient accuracy for engineering and scientific calculations. The Differential Analyzer could only solve those problems which it was wired to solve - it could not change the direction of information flow without a prior wiring change. The second of these devices was Dr. Howard Aiken's Mark I Automatic Sequence Controlled Calculator, built at Harvard University with support from the International Business Machines Corporation. This machine used punched card equipment for data storage and telephone relays as switching elements, and received its instructions from punched paper tape. It was the first successful large scale digital computer and was about ten times faster than the desk calculators of that day. It was at this time that the word computer took on its present connotation. Both of these devices could accept data and instructions from a wiring set up (in the analog) and from punched cards and paper tape (in the digital) and proceed through a problem, from beginning to recorded solution, without assistance by an operator. The ability to store significant amounts of data and instructions and to work without human intervention is the distinguishing hallmark of the computer as opposed to the calculator or other computing devices.

In 1946, Dr. J Presper Eckert and Dr. John W. Mauchly completed work on their Electronic Numerical Integrator and Calculator (ENIAC) at the Moore School of Engineering of the University of Pennsylvania. Their digital computer, sponsored by the Ordinance Department of the Army, was the first to utilize the vacuum tube in place of the slow relay for switching. The local tabloids suggested that the lights of Philadelphia would dim when the ENIAC was turned on because of the power consumption of the tens of thousands of vacuum tubes. Several learned critics suggested that the starting voltage transients would always blow at least one tube and, as a consequence, the machine could not work. While these fears proved to be not totally correct, it is interesting to note that early computers of the ENIAC class were operated continuously and powered down only for emergencies. Later machines in both the analog and the digital fields used vacuum tube circuits, then transistor circuits and now integrated circuits in an effort to gain computational speed. Today general purpose computers of both types are electronic; the electro-mechanical types of the 1940s are obsolete.[3]

---

[3] Material in this and following sections has been adopted from R. C. Heterick and J. H. Sword, *An Introduction to Computers and Elementary FORTRAN* (Dubuque: Kendall-Hunt Publishers, 1966.)

The first commercial model of the computer was introduced in 1951. The first computer was delivered to a privately owned company (the General Electric Company) in 1954. By that time the experts had revised their previous figures upward and were predicting that possibly 50 United States corporations would eventually acquire digital computers. Ten years later one large corporation (Westinghouse) was reported by FORTUNE Magazine to have spent more than $16 million in one year on computer rental, research and salaries of computer personnel. The company reported a return of more than $20 million on its investment. Similar experience by others increased the popularity of the computer until (at this writing) the computer industry is a multi-billion dollar per year business, with one manufacturer alone accounting for over $10,000,000,000 worth of equipment manufactured per year.

There are, in operation at this time, more than 500,000 computers (exclusive of microcomputers) manufactured by United States manufacturers alone. Practically all of these computers have been manufactured since 1964 and all of them are faster, less expensive (in terms of the number of computations per unit of cost) and have more storage capacity than the first electronic computer. Fewer than ten percent of them rent for as little as $1000 per month and rentals as high as $500,000 per month are predicted for systems currently under development. Computer rentals in excess of $50,000 per month are commonplace. The majority of the large scale computer systems are used by government and private research laboratories and by the manufacturing and process industries.

While we tend to think of computers as large complex machines we fail to realize just how many small (desk top to chip size) computers are commonly in use. Computers on a chip for less than $100 are available. The instruction repertoire and storage of these micro-processors compares favorably to first generation equipment. The fastest growing segment of the computer industry is the mini- and micro-processing field. Predictions have been made that the annual sale of micro-processors in 1985 will exceed the total production of computers for the previous 30 years.

Although the United States is the world leader in the manufacture of computers, there are computer industries in England, France, Germany, Italy, Japan, Russia and Sweden. Still the demand for computers rises.

Computer development was concentrated in university research labs during the five-year period between the completion of the ENIAC and the introduction of the first commercial computer, the UNIVAC I. The stored program concept was exploited and index registers developed. Out of this transitional period came the EDVAC, the ILLIAC, the MANIAC and the JOHNIAC following the basic format of the Eckert and Mauchly work and the Whirlwind, Zephyr and Hurricane building upon the work of Aiken. Concurrent efforts in large scale analog devices, the Cyclone and the Typhoon, were undertaken by industry.

First installation of the UNIVAC was made in June, 1951 and was priced at $250,000. The first order for the UNIVAC was, interestingly enough, placed by the Bureau of the Census. For nearly five years the UNIVAC essentially had the electronic computer field to itself. As late as 1948 the IBM Corporation had made the decision not to market the computer because they felt it would never be profitable. The corporate effort to develop the 700 series of IBM computers and the obvious success of that effort will rank as one of the major business success stories in history.

This first generation of computers was characterized by vacuum tube memories, tape storage and punched card input output devices. The mean time between component failures was frequently no more than 30 minutes. These first generation computers were generally programmed in machine or assembly language. Control of the computer system was vested in an operator who usually was the programmer.

The second generation is usually marked by the switch from tube to transitors and magnetic core memories. Computer memories were increased from 2,000 to 30,000 words. A million operations per second was achieved in the second generation, as contrasted with ten per second on the MARK I, 5,000 per second on the ENIAC and 100,000 per second in the first generation. During the second generation computers came to be viewed as something more than a scientific tool and several computers were designed specifically for commercial applications. The second generation boosted the number of operation codes from 30 to over 100 and ushered in the era of time sharing, albeit in a rather jerrybuilt, crude fashion. Procedure languages such as FORTRAN and COBOL became the primary programming tools and more advanced problem-oriented languages such as COGO and STRESS were developed. Rudimentary operating systems were developed and multi-processor configurations were implemented to overcome the speed differential between the electro-mechanical I/O devices and the totally electronic central processor.

The third generation is usually marked by the introduction of the IBM 360 line of computers. Third generation computers typically employed monolithic integrated circuits rather than transistors and were characterized by system control programs called operating systems. The third generation marked the general acceptance of time sharing, first in the form of remote printer card reader batch complexes, and later as slow speed (ten-character-per-second) interactive systems communicating over standard telephone lines with teletype-like devices. Significant experimentation took place with large data bases featuring online inquiry and update. Third generation technology led to attempts to consolidate computing activities, generally expressed as the economy of scale phenomena. The cost per unit of storage and cost per instruction cycle time were considerably less for the larger machines. Dr. Herbert Grosch was one of the first to observe this phenomena and suggested that a doubling of the price of a computer tended to quadruple its speed. The third generation marked the transition from hardware centered computing to our current software development focus. The impact of this transition on organizational structure, software development and operations management has been major and is pursued in detail in subsequent sections. This transition has not been fully absorbed by many computing organizations and continues to be the source of a myriad of development and management problems.

The depersonalization of computing in the third generation, accompanied by significant advances in large scale circuit integration and telecommunications technology, ushered in what will probably become known as the fourth (and last?) generation. For our purpose, we will mark the fourth generation by the proliferation and acceptance of the mini- and micro-computers and significant efforts toward distributed computing in networks. The mini-computers of the 1970s have roughly the same power and storage capacity of first generation computers at a tenth or less of the cost. In fact, some of the pocket size calculators compare favorably with the capabilities of first generation computers. Hierarchical networks of computers featuring widespread use of cathode ray tube input devices are probably the hallmark of fourth generation systems. Sensor based applications, wand readers, environmental automation, electronic fund transfer, etc., are becoming commonplace. The computer has probably reached the level of the ultimate ubiquitous machine.

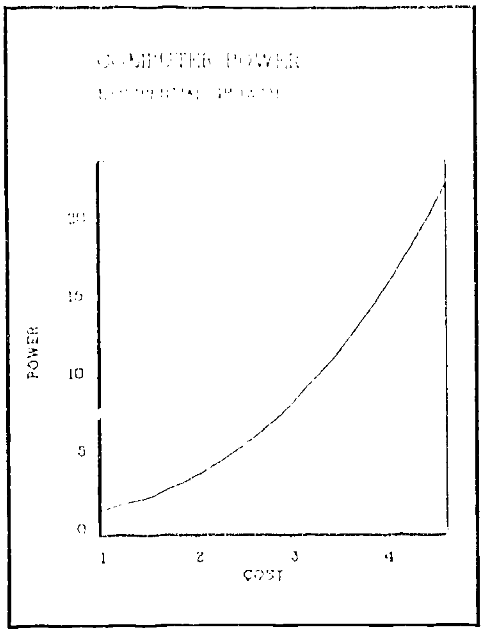



Figure 9
Grosch's Law:
power increases
exponentially
with cost

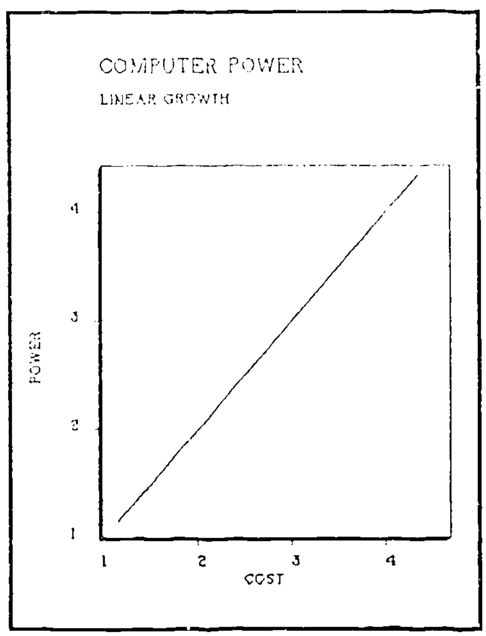
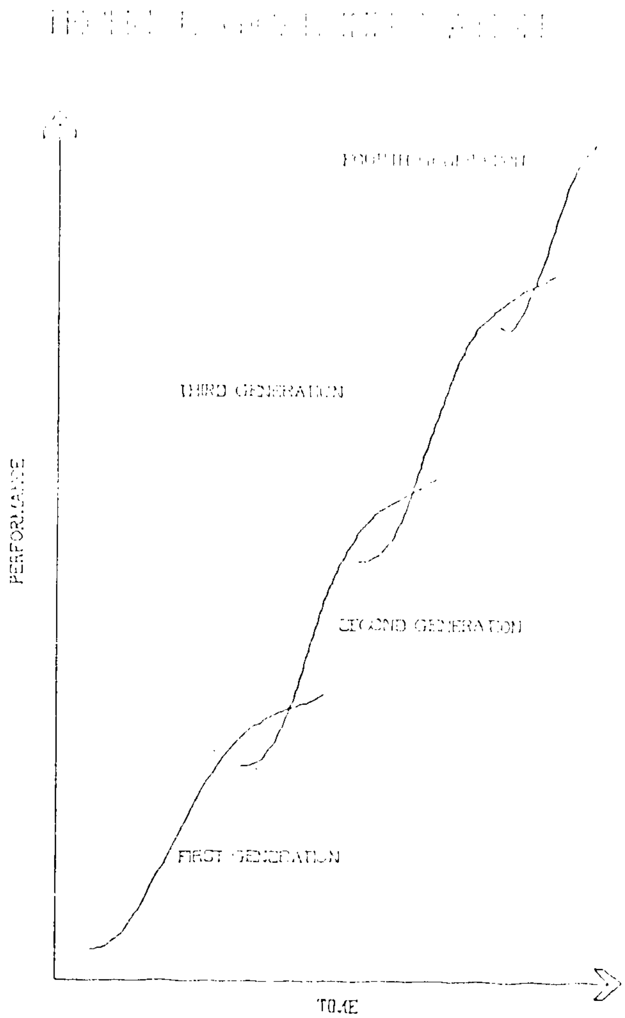



Figure 10
Micro Phenomena:
power increases
directly with
cost




THE IMPACT OF TECHNOLOGICAL BREAKTHROUGHS
ON THE PERFORMANCE OF EQUIPMENT

Figure 11

## Architecture of Computing Systems

The digital computer is, by definition, a counting device. Its manner of counting is not the same as that of its users, however. After first building computers using the familiar decimal (base 10) system, early computer designers recognized the advantages inherent in the binary (base 2) number system. So many of the devices used to store or transmit data and numerical instructions are of the two-state or bi-stable type it was inevitable that computer designers would turn to the binary system. A switch, for example, is either on or off; a card either has a hole punched in a given position or it does not; the flux lines around a magnetic core are either clockwise or counter-clockwise; and so on. If we use the numerals 0 and 1 to represent the two possible states of a binary device and arrange them in rows such that the presence of a 0 in any position means zero in the decimal sense and the presence of a 1 means 2*n where n is the position number in a given row, we may represent decimal numbers in terms of their binary equivalents.

Such groupings of symbols, or assemblages of two-state devices, are used as registers in the computer and the contents of such registers may be combined to perform computations.

Rules for performing the various arithmetic operations exist for binary number systems just as they do for other systems. The average computer user is spared the task of learning these rules, however, because he communicates with the computer using his own language. He may, for example, submit his original instructions and data using the alphabet, standard punctuation symbols and the familiar decimal system for numerical values. The translation to binary form is made by machine. Results or other information going from the computer to the user are translated from binary and returned to the user in a form which he is accustomed to reading.

The academic administrator will most likely use the computer as a tool for solving problems related to his special field of interest. He really does not need a detailed knowledge of the design or operation of the various parts which make up a computer system. The user should, however, have some general understanding of the flow of information through the computer. The block diagram or five-part black box representation has been widely used to explain the operation of the computer. User attention has been focused, and rightly so, on the input/output processes of the model. However, technological developments in processor design have a significant impact on the economics of the input output operations. A subsequent section will investigate several of these new technologies so that a basis for later discussion of changing economics may be established.

## Control Unit

The control unit is the dispatcher of the system. All instructions, data and operation commands are routed through this device and it controls the operation of all internal and external equipment attached to the system. The operation commands for all such equipment are numeric in nature and are transmitted and received as electrical pulses. The control unit is not directly accessible to the programmer and he or she generally has no special interest in it so long it functions properly.

## Arithmetic and Logic Unit

The arithmetic capability of the machine is, in effect, limited to addition, subtraction, multiplication and division (some new scientific computers have square root capability and other unique operators). The logical capacity is that of distinguishing between zero and non-zero and between plus and minus which, when coupled with the subtraction operation, gives the added ability of determining whether one number is different from another and, if so, which is larger. The arithmetic unit also has the capability of shifting numbers to the right or left in its registers.

Both instructions and data are numeric in form and the arithmetic unit may modify either or both of them upon command. Also, since alphabetic quantities must be represented in numeric fashion, they, too, may be manipulated by the arithmetic unit. Such operations are binary in nature and they can be accomplished in times measured in microseconds (millionths) or less in modern computers.

The smallest unique unit in a computer is called a bit (Binary digIT). For addressing purposes, bits are usually grouped into bytes (typically eight bits) or words (typically 16 to 64 bits). If an eight-bit byte is used for character representation, 256 unique characters (2*8) can be represented. This easily provides for upper and lower case alphabets, the decimal numerals, special characters such as +, -, $, etc. and other special characters such as those used in APL (A Programming Language). National and international standards groups exist to bring some order out of the confusion of many different code representations. These differing code representations are one of the first problems which must be overcome in creating a network of dissimilar computers.

Memory (Storage Unit)

Basic working storage in all modern computers is of the magnetic type and usually consists of a three-dimensional array of small ferrite rings called cores or some form of monolithic integrated circuitry. In the core type each ring has a unique address and can be magnetized by the passage of a current along each of two wires which go through the center of the ring. By appropriate selection of wires and by control of the direction of current flow, any given core element may be addressed and set with a binary 1 or 0. The individual cores (or basic storage units for the non-core types) is a binary bit. In some machines the computational unit is a word; other machines permit words to contain a variable number of bits. In either case, the word may be located by means of a specific and unique numeric address which is analogous to a street address or post office box number. Still other types of computers use an addressable basic unit of the byte; words in these machines are generally made up of two to eight bytes. Access times are extremely fast for modern computers; for some of them, times are measured in nanoseconds (billionths), for others microseconds.

Both data and instructions are stored in a binary form in memory. Storage in memory is considered permanent in that a word stored in a given address will remain there until it is actively replaced by another store operation. Interrogation of a storage address or the print-out of information from it will not destroy the contents. The read-in of information or the redefining of a value already in storage always destroys the previous contents of a storage address.

Auxiliary storage may exist in magnetic tape units, magnetic drums, magnetic disks, punched tape or punched cards. Units equipped with optical scanning devices may, in addition, utilize printed storage. Access time for such units is generally measured in milliseconds (thousandths) or slower. With the magnetic devices, however, once the machine has found a piece of data, the transfer rate from auxiliary storage to working storage is quite rapid. These devices may be large enough to store practically limitless amounts of information (the real constraint is financial which, of course, must be based upon need and frequency of use).

Much research effort is being placed on developing larger and faster computer storages. There are currently storages available which will store a trillion bits (terabit stores) but with rather slow access times. Thin film and bubble technology is becoming quite advanced. The laser holds perhaps the greatest potential for extremely large, fast computer stores.

Input Unit

The programmer communicates with the computer through the input unit. This unit usually is a typewriter or punched tape reader on very small computers (or a teletype like unit in remote terminals) and a punched card reader on medium and large scale systems. There may be intermediate input devices, also. Larger systems usually permit programmer access only through magnetic tape or disk. The programmer has this punched card information read by a small computer and stored on magnetic tape or disk for transmittal to the large computer in such systems. There is at present a growing market for recently introduced keyboard-to-computer systems which bypass the punched card operation altogether.

Regardless of the type of input, the input unit senses coded information of some type, translates it, and sends it to the computer as a string of timed electric pulses. It should be noted that most input devices do not have control functions independent of the computer except for operator controlled START and STOP. Punched cards, in particular, cannot be recalled for reading again within a given problem solution or program.

Operating speeds vary greatly for the various types of input devices. The slowest is, of course, the type which depends upon manual operation the typewriter or the teletype unit. The fastest of the types mentioned above are the magnetic disk and drum.

The primary characteristics of both input and output units is their speed differential as compared to the Central Processing Unit. While the CPU is capable of billions of operations per second, typical I/O units operate at ten characters per second, 300 cards per minute or 1000 lines per minute. So long as these devices are electro-mechanical or actuated by humans, this speed differential will continue to exist.

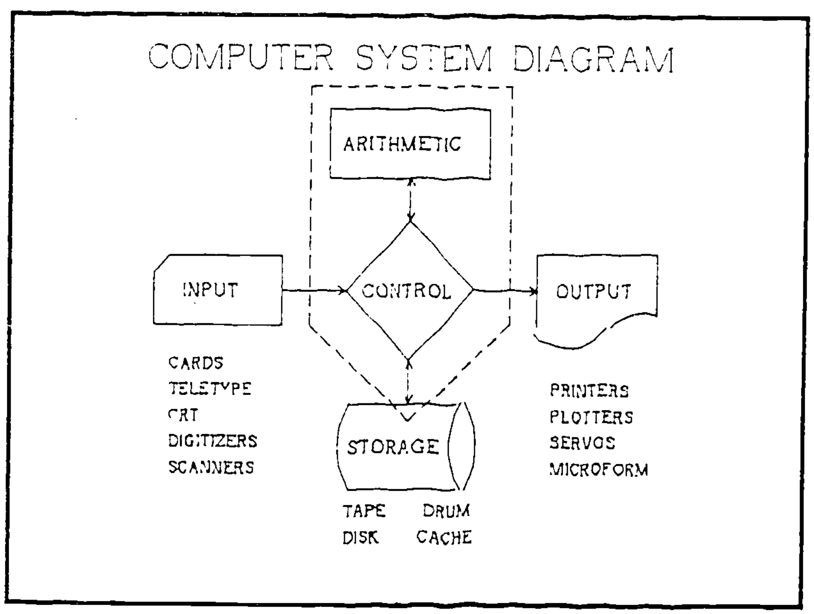

COMPUTER SYSTEM DIAGRAM
ARITHMETIC
INPUT
CONTROL
OUTPUT
CARDS
TELETYPE
CRT
DIGITIZERS
SCANNERS
STORAGE
PRINTERS
PLOTTERS
SERVOS
MICROFORM
TAPE
DRUM
DISK
CACHE

Figure 12

Output Unit

The computer transmits information and results to the programmer by means of magnetic tape, punched paper tape or cards, or the output may go directly to a printer, a plotter, a CRT or some other device. Regardless of the type of output device, information will ultimately go to the user in one of the languages with which he is familiar. For the academic administrator this will usually be in numeric, alphabetic and/or graphical terms. Again, speeds are quite variable.

Computer Languages

Each computer is designed to accept certain combinations of electrical pulses which can be interpreted as numerical characters. The computer is essentially a symbol manipulating machine. It does not matter at all to the computer what the binary digits represent so long as man can develop the necessary systems of unambiguous symbols. The code for a machine, consisting of symbols recognizable to the programmer and which, when read from punched cards or transmitted from a terminal, create the necessary electrical pulse chains, is called a language. Languages exist at four levels and are classified as follows.

Machine Language

The basic combination of codes accepted by a given machine is unique. This combination, written by the programmer in numeric form, is called machine language. Programming in machine language is very exacting and time-consuming. Such programs require a separate step for each computer operation and programming steps must be in the order of execution. The programmer must specify exact locations in memory for each separate piece of information and must take care not to use the same location for two separate values.

As an example of machine language programming, consider the following system (used with a popular machine of several years ago). Instructions to the machine were of the form:

| OPERATION CODE | DATA ADDRESS | ADDRESS OF NEXT INSTRUCTION |
|---|---|---|

Solving such a simple problem as X=A+B-C required a sequence of steps as follows (where 60 is the code for clearing a register and adding to it, 65 means add without clearing, 70 means subtract without clearing, 80 means store in memory and the four-digit groups are addresses of storage locations):

| LOCATION | INSTRUCTION | | |
|---|---|---|---|
| 0490 | 60 | 2050 | 0500 |
| 0500 | 65 | 2060 | 0510 |
| 0510 | 70 | 2070 | 0520 |
| 0520 | 80 | 2080 | 0530 |
| | .. | .... | etc. |

Before the program could be executed, the programmer had to store the program steps and data in the assigned addresses (the second operation in storage location 0500, the third in location 0510, etc.) and send the machine to the first step (location 0490) by means of switch settings on the machine console. (Note that the small program segment does not contain instructions for obtaining the result of the calculations, this would require additional instructions.)

Needless to say, such programming techniques did not prove to be popular with problem solvers. Today, this type of programming is the province of the professional programmer who is involved in computer development or in development of languages.

Assembly Language

Because of the difficulty which programmers had with bookkeeping operations, assembly languages which accepted symbolic addresses and operation commands were developed. Machine language programs called assemblers, written by professional programmers, cause the computer to accept symbolic names for numerical addresses, convert those names to addresses and to build an index of locations used. The assembler is also designed to accept mnemonic operation commands. The programs have to be translated to machine language for execution. The program submitted by the programmer is called a source program; the instructions generated by the computer constitute an object program. Such programs still require one instruction per computer operation. Although not as time-consuming to the programmer as machine language, such programming is tedious and generally less efficient in its exercise of machine resources.

Using an assembly language the example problem above might be written (where CLA means clear and add, etc.):

```
CLA A
ADD B
SUB C
STO X
```

Note that the assembly language has the same basic structure as the machine language which it replaces and is therefore machine dependent. This, too, is unpopular with problem solvers because they are not accustomed to breaking their problems into such minute operations.

There is one procedure which has been popular with those who must program in assembly languages. Certain groups of operations which frequently occur together can be grouped and the entire group called to step-by-step execution by a single command. Such an instruction is called a macro-assembly instruction. The use of such commands was the forerunner of the procedure language.

Procedure Language

There have been many attempts to make programming easier by permitting the programmer to write in a language related to some method or procedure with which he was already familiar. The FORTRAN (FORmula TRANslation) language is perhaps the most successful of those attempts; certainly it is the most widely used. The programmer writes a source program in the FORTRAN language and submits to the computer first a machine language program called a compiler and then the source program. The machine, under the direction of the compiler, will either translate (or compile) the source program into a machine language program (called an object program) or into an assembly language which is then translated. The program is then executable.

The advantage of this approach is that one FORTRAN statement can represent many machine language steps. Our previous example would appear in a FORTRAN program as X=A+B-C There is a penalty in increased computer time required but that penalty is more than offset by the savings of programmer time and the ease with which the language can be learned. It is generally agreed that the procedure language is best for beginning programmers; indeed, it is adequate for the needs of most professional programmers.

Procedure languages offer the added advantage of being machine independent. The compiler which translates the source program is machine dependent, of course, but the source language may be used to write programs for the machines of several different manufacturers. Minor variations occur but, generally speaking, FORTRAN type programs of a given level are compatible between machines which operate at that level.

Problem Language

In certain specialized areas where everyone working on a particular type of problem is likely to use the same standard approach, languages have been written in which a particular symbolic name may call forth a complete sequence of operations (analogous to solving a group of equations or FORTRAN type statements) and deliver results in some predefined format. The structure and terminology of such languages is dictated by the type of problem being solved. They are called problem oriented languages and are usually written using a procedure type language such as FORTRAN.

Examples of such languages may be found in many different problem areas. There are, for example, a STRUctural Design Language called STRUDL; an Electronic Circuit Analysis Program called ECAP; a COordinate GeOmetry program called COGO; a discrete system simulation program called G SS and many others.

## Modes of Computing

Three principal modes of computer usage are usually identified. They are batch, interactive and real time. Batch systems were the rule during the first three generations but are rapidly being supplanted by interactive and real time applications in the fourth. Most procedure languages were developed for execution on batch systems, APL and BASIC being the two dominant exceptions.

In a batch system the programmer prepares the source program and data, typically on punched cards, and deposits them at a pickup station. The cards are read into a computer system and temporarily stored on disk or tape, a process sometimes termed spooling. A job processing scheduling algorithm, part of the computer's operating system, reviews the spool queue and selects jobs for processing.

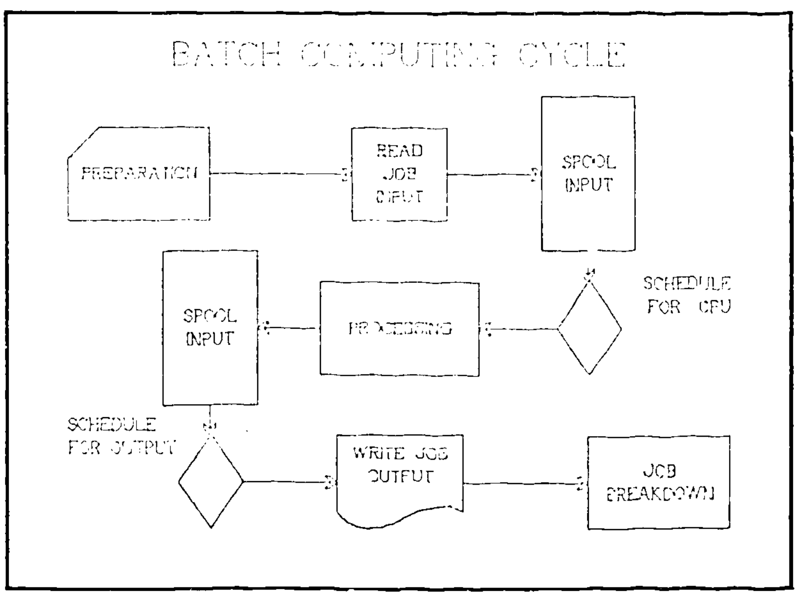

BATCH COMPUTING CYCLE
PREPARATION
SPOOL INPUT
SCHEDULE FOR CPU
SPOOL INPUT
PROCESSING
SCHEDULE FOR OUTPUT
WRITE JOB OUTPUT
JOB BREAKDOWN

Figure 13

When the job has been selected and processed it is again placed on disk to await scheduling for output to a printer, plotter, card punch or other output media. A job output scheduling algorithm searches this spool queue and eventually selects the job's output for processing. Typical turn around times for a job may be from three to twelve hours. The batch operation is essentially a four-step process. While this is an efficient and reasonable method for processing production jobs it tends to be quite inefficient of the programmer's time when new code is being written and tested, particularly when the job aborts due to a key-punching or trivial programming error and the programmer must wait over three hours to discover and correct the error.

In order to overcome these delays during the program development phase, time sharing was developed. Time sharing allows the user an interactive mode of computer usage, permitting many simultaneous users to feel as though they each had control of the computer. This effect is possible when you consider the speed differential between the computer's processing ability and the reaction time of the programmer typing at a keyboard. Well designed interactive languages will scan input lines for syntax errors, allowing immediate correction of typing and trivial programming mistakes. A well designed interactive language will also facilitate a desk calculator mode, permitting many operations without the necessity of coding a multi-step program. In an interactive system a dispatching algorithm looks through the queue of terminals connected to see which is ready for service and gives some small quantum of time to each in a round robin fashion. Several hundred terminals could be connected, each having the appearance of controlling the computer. Typical response times for an interactive system would be in the one-to-five second range. While not as efficient in utilization of machine resources as batch, programmer productivity is greatly increased (probably at least ten-to-one) over batch systems. The interactive environment leads immediately to thoughts of electronic message switching and office automation. A subsequent section will discuss text editing and transmission in greater detail.

Real time systems were developed for military command and control applications, the SAGE (Semi-Automatic Ground Environment) system for ballistic missile warning being implemented in the 1950s. The space program made heavy use of real time systems. Among the many applications was vehicle guidance, which required the results of over 15,000 serial computations to calculate and make flight path corrections. Obviously some form of round robin computer sharing such as in interactive systems just would not do. The characteristic of real time applications is that they are so important that they must take precedence over all other uses of the computer. Many forms of process automation currently rely upon real time computer systems. Most currently designed rapid transit systems require some form of real time computer control.

As computers were accepted and proliferated, it was only logical that attempts would be made to link them into networks. The examples of currently existing power grids and telephone systems were a persuasive impetus. Particularly in research and education, programming was somewhat of a cottage industry. How to use the code written by someone at another computer site has been a problem for computer users since the days of UNIVAC I. Additionally, some form of networking would allow local specialization and the advantages of a market economy in computer services. The development of time sharing provided the majority of the telecommunications technology necessary to create networks of computers above networks of terminals. Some techniques needed to be developed to overcome the error rate in conventional telephone service, about two errors in every 10*5 bits transmitted. The currently favored approach is packet switching where data is sent between computers in small packets of about 1000 bits. These packets may be sent between computers by several different routes, each intermediate computer storing, checking, then forwarding the packet on through the network, with erroneous packets simply being re-transmitted. Upon arrival at the final destination the packets are re-assembled, placed in proper sequential order and presented to the target computer system. All of this network activity requires logic which is provided by small computers acting as front ends to the major computer systems in the network.

Most research in computer networks has been concentrated in communications technology. In order to get higher transmission speeds at reduced costs digital communications companies have been chartered in competition with the Bell System's analog network. In response to these initiatives the Bell System is implementing Data Under Voice (DUV) service utilizing under-loaded voice lines to pass digital information. Typical transmission rates vary from 2000 bits/second (ordinary telephone service) to 50,000 bits/second. Computer data has been sent by laser, microwave, and increasingly by satellite. Several large suppliers of computer services currently have their own satellites in orbit. Even more exotic technology such as millimeter wave lengths and helical wave guide appear in the offing.[4] High density communications channels can be shared or multiplexed between several users by time division (much as the large oil pipelines are shared) or frequency division (a 2000 bps line may be shared by about six 300 bps users).

The potential of computer networks has led to speculation about local and national computer and information utilities. One major computer network, the ARPA (Advanced Research Projects Agency of the Department of Defense), already links over 40 major universities and research laboratories on the mainland via telephone circuits and Hawaii via the ALOHA satellite. We can expect to see continuing growth of EDUNET and the further development of local and regional resource sharing networks in higher education. The development and widespread use of networking technology should make the use of computers in academic administration extremely cost effective.

---

[4] For an imaginative look at the future see James Martin, *Future Developments in Telecommunications* (New York: Prentice-Hall, Inc. 1971).

# 2

# TECHNOLOGICAL TRENDS

"Every problem contains within itself
the seeds to its own solution."

## Cost of Computing

The purpose of this chapter is to examine some of the economic and technological influences under which computing centers of the 80s will have to operate.[5] The Fall 1979 *EDUCOM Bulletin* contains an article that projects the cost of computing in the next decade under the following simple assumptions:

- Hardware costs will continue to decrease at the rate of 20% a year

- Software and personnel costs will continue to increase at 8% a year

Under these assumptions we may conclude that for no growth computing, by 1989 the cost of hardware will be approximately 5% of the total cost of computing. This

----------------

[5] This chapter appeared almost in its entirety in *CAUSE/EFFECT*, November 1981, pp. 12-19.

means that approximately 95% of total computing costs will be attributed to software and personnel. Projected costs for software and personnel show an even more dramatic increase and consume a greater percentage of the budget if a 15% annual growth is assumed (see Figures 14 and 15).[6]

The purpose of this section is twofold: first to determine if the above assumptions and conclusions are valid; and second, to outline management strategies that should be used to cope with and to take advantage of the changing situation.

## Hardware Costs

The previous chapter discussed the major components that make up the computer system. A large portion of the total cost of computer hardware can be attributed to two components: the central processor unit (CPU) and the memory (including main memory and secondary storage). Thus if computers are to continue to drop in prices, these price drops must come from lower costs for CPU and memory. Lower costs can be achieved either through improved performance of the component or through reduced unit manufacturing costs.

### Potential for CPUs

Let us examine the CPU performance issue. One measure of the speed of the computer is the machine cycle time. Cycle time is the time interval in which a computer can perform a given number of operations. A large commercial computer today has a cycle time varying between 12 and 80 nanoseconds (a nanosecond is 1,000,000,000th fraction of a second). The fastest computer today has a cycle time of approximately five nanoseconds. A 20% price drop over the next decade will require a 50 fold improvement in performance; i.e. computers with a cycle time of 1/10 nanosecond would be required. Is it possible to produce such a machine or have we reached the limits of the technology?

---------------

[6] Douglas E. Van Houweling, "Meeting the Challenge of Diversity and Change: Cornell Computing Services," *EDUCOM Bulletin*, Fall 1979, p. 5.

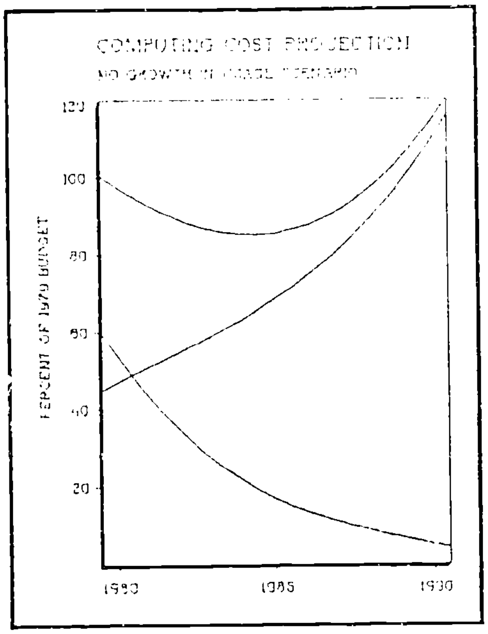



Figure 14

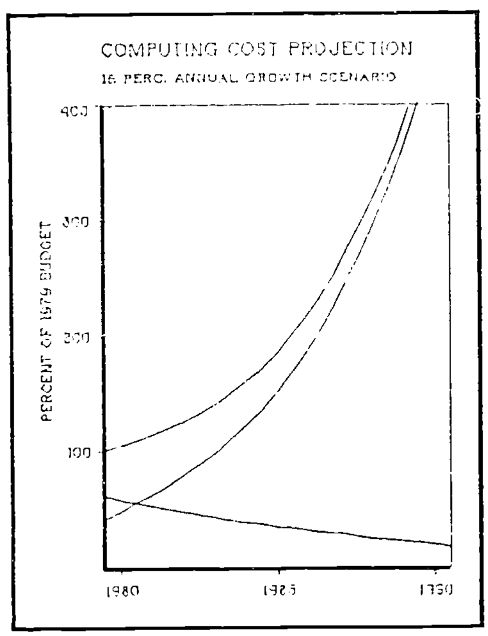



Figure 15

Lewis M. Branscomb examines this question from the perspective of physics.[7] He argues that the speed of light is such that it travels about three centimeters in 1/10 of a nanosecond. Thus computers with a cycle time of 1/10 nanosecond cannot be larger than say a one-inch cube. This is encouraging in the sense that the smaller the package the lower its requirements for power and the cheaper its production costs. If the computer has 300,000 circuits then it will generate about one kilowatt of heat. Today's technology will not permit us to extract one kilowatt of heat from one inch cube space without damage. Thus it is likely that semiconductor technology will not be the answer to the requirements that we might have towards the end of the decade. However, a new technology, called Josephson Junction technology (named after Nobel Prize winner Brian Josephson) holds a lot of promise. Josephson junctions use metal alloys cooled in liquid helium to 4.2 degrees above absolute zero. At this temperature the metals become superconducting and lose their resistance to electricity. Circuits built with this technology have switching speeds in excess of 20 trillionths of a second. Thus Josephson junction technology, presently in the research labs, holds promise for the future, but much needs to be done before the junctions become a viable commercial product.

The silicon-based technology that we are using today is by no means dead, as substantial improvements are still possible. Todays technology allows around 30,000 transistors on a chip. It is estimated that half a million transistors can be put on a chip by 1985 and about ten million by 1990.[8] Full computers on a chip should be a reality by the mid-1980s.

Another cost-saving approach in the manufacture of computers is the use of gate arrays. Large-scale integrated circuits are made on silicon wafers. The circuitry is fabricated on the wafers by a multiple step process called masking (typically a completed wafer has seven or eight masks). The gate array is a partially complete wafer that awaits the customer's logic requirements. The user simply provides the logic information for the last two masks. This process reduces the overall cost of the wafer and also hastens the design of the chip. Extensive use of gate arrays

---------------

[7] Lewis M. Branscomb, "Information: the Ultimate Frontier," *SCIENCE* 203 (12 January 1979): 144.

[8] Edward K. Yasaki, "Markets," *Datamation* (Special Edition), December 1980, p. 10.

can possibly cut the cost of the system by a factor of four. It is, therefore, not unreasonable to expect that silicon based technology will continue to provide substantial cost reductions for at least five years, perhaps longer.

We have already seen that CPUs in computers are very fast with cycle times as low as five nanoseconds and getting faster. What is not widely known is that they are very consistent with regard to error rate, already as low as one in one trillion. If Josephson junction technology becomes widely used, it will be safe to say that the computers of tomorrow will be very cold indeed. Thus computers may be characterized as being fast, consistent, cold and dumb. On the other hand humans are slow, inconsistent, warm and brilliant. The real challenge is to get the two working together using the optimal mix of the capabilities of the computer and the talents of the human.

Memory Systems

Memory systems are keeping pace with the developments in the CPU area. Main memory prices have shown significant drop with the introduction of 4K, then 16K chips. The recent price drops in the 16K chip market indicate that perhaps 64K chips are ready for use, with 256K memory chips available in the not too distant future. If this is true, it comes well in advance of projected time frames. The larger density chips are both less expensive and more reliable.

The developments in the area of secondary storage have been more dramatic. The first magnetic recorder dates back to the year 1900 when Danish engineer Valdemar Paulsen demonstrated a device in the Paris Exposition.[9] The invention of the vacuum tube amplifier in 1920 gave a big boost to the weak signals from the magnetic recorders, and in 1923, Thomas Edison built the phonograph. Whereas the main memory of the computer is based on semiconductor technology, the secondary memory, so far, is based primarily on magnetic disks. In magnetic disks, the reading and writing of data is accomplished by a read/write head which is an electromagnetic device that resembles the head of a home tape recorder. The head positions itself on a specific position on a rotating disk made of magnetic material. Each head position corresponds to a concentric circle on the disk called a track. Data are read from and written on tracks.

---

[9] Robert M. White, "Disk Storage Technology," *Scientific American*, August 1980, pp. 138-148.

The most commonly used technology for disk drives is called the Winchester technology, introduced by IBM in 1973 for their 3340 disk drives. In this technology the read/write head consists of three rails. The outer two rails govern the flow of air and allow the head to float 17 microinches above the surface of the disk. By comparison a smoke particle is 100 microinches in diameter and a human hair is 2500 microinches. The tail end of the center rail has the read/write head built on it.

The amount of data that can be stored on a disk is determined by number of tracks per inch of radius of the disk and by the amount of data that can be stored per inch along the track. The Winchester head allows for 1000 tracks per inch with a recording density of 10,000 bits per inch along the track. A Winchester disk is between eight and 14 inches in diameter and is contained in a sealed unit to keep out dust particles.

The material on the surface of the magnetic disks is iron oxide. Tests show that if metallic cobalt were used instead of iron oxide, greater storage densities could be achieved. Another area of improvement is the read/write head itself. The new 3370 disk drives introduced by IBM use a thin film head. This head is made of permalloy, a mixture of nickel and iron. Unlike the Winchester head it has no coil of wire in its electromagnet; it uses a spiral film of electrical conductor to form the electromagnet. Compared to the 10,000 bits per inch for the Winchester technology, the film-head can record at a density of 15,000 bits per inch today and has a potential of going up to 25,000 bits per inch in the future.

There is yet another possibility of improving the recording density on the disk drives. At present, data are recorded along the surface of the disk. It has been suggested that data could be stored on end, perpendicular to the surface of the disk. The technology to do this is yet to be discovered. Both a suitable magnetic medium and a read/write head capable of doing this will have to be developed. If this can be done, improvements in recording density are possible by a factor of ten.

Optical Disks

A relatively new medium for computer data storage is the optical system. Conceptually, optical systems are very similar to magnetic systems in that both require disk medium for storing data and read/write heads for storing and retrieving data. Both systems require similar mechanisms to position the heads on tracks and the necessary logic to code and decode data suitable for storage and error recovery. Apart from the fact that one uses electromagnetism and the other uses laser optics, the important difference between the two systems is that in optical systems the energy used to detect the presence or absence of a "bit" is external to the storage medium. The storage medium the disk simply acts as a gate. This allows for much greater storage densities. The simplest form of optical disk is one in which data is stored on the surface of the disk by a laser beam burning miniscule holes on the surface. The storage density is extremely high about 100,000 bits per inch. A standard 14 inch disk is capable of storing approximately 10 billion bits of information. Clearly, though, the storage of data using this technique is non-erasable.

Optical disk devices are rapidly becoming a mass consumer product. Because of the high volume that is possible in consumer products, the costs of optical disks for data storage can have an improvement of a factor of 10,000 over conventional magnetic media. Further, the entire recording medium is removable allowing for the flexibility of removable disks or tapes.

Efforts to develop an erasable optical medium that can be reused (i.e., updatable) have, so far, been unsuccessful. Optical disks hold the potential of improving price performance by a factor much greater than 100 and are thus well within the 20% a year (for the whole decade) criteria we are examining. The magnetic medium is not dead by any means. The projected improvements in the conventional magnetic disk technology will continue to sustain, for several years, the price performance figures that we are seeking.

Bubble Memory

Bubble memory will play an important role in the storage and retrieval of computer data. What exactly are magnetic bubbles? For our purposes, it will suffice to say that magnetic bubbles are very small and cylindrical magnetic regions about 1/25,000th inch in diameter. These so called "bubbles" are found to have a stable existence in certain materials like orthoferrite or garnet. Further, the bubbles can be made to move along the surface of material by the application of a proper magnetic field. The stability of the bubble and its controlled movement along the surface of the storage material forms the basis of bubble memory. (An excellent exposition of magnetic bubbles can be found in the August 1980 edition of *Scientific American*).

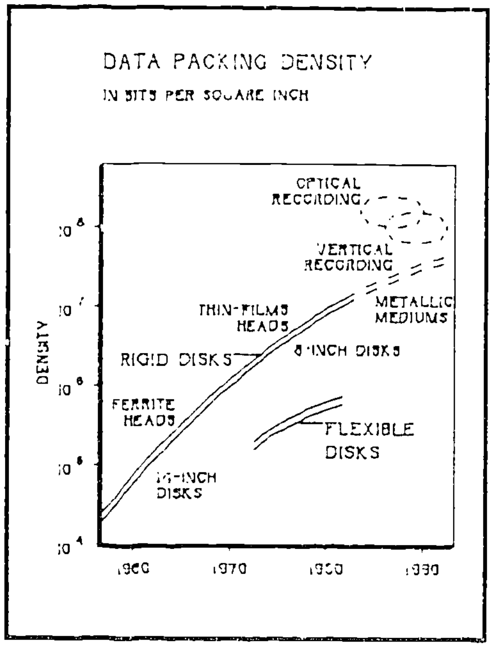



Figure 16

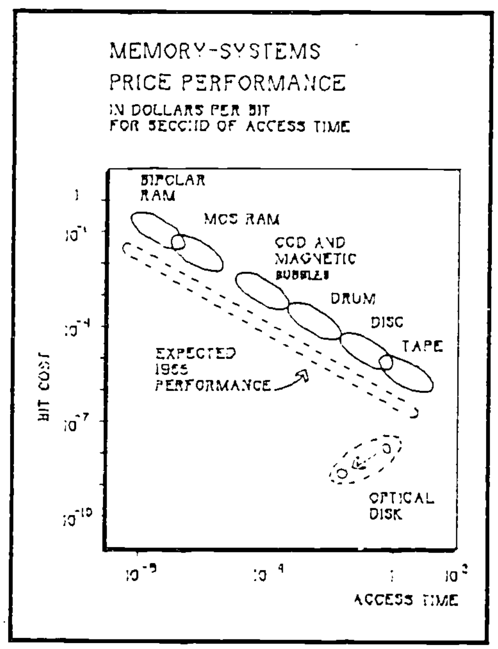



Figure 17

The great advantage of bubble memory over disk memory is that it has no moving parts. Thus, unlike mechanical devices, it requires no maintenance. It is estimated that bubble memory has a capacity of retaining data over 100 years. This long term stability of data compares very favorably with the estimated life of data on magnetic tapes. The error rate for magnetic bubble memory is about one in 100 trillion bits; thus bubble memory has an error rate two orders of magnitude better than CPUs. Compared to disk drives bubble memory has much higher data densities and also has superior access times. The only disadvantage of bubble memory is a slower data transfer rates.

The price performance of bubble memory is very attractive; 100 bits of data can be stored for about one cent. Figure 16 shows the location of bubble memory in the price performance graph. Clearly, bubble memorys fill a rather large gap between main memory and disk memory. The two very important features of bubble memories - low maintenance and long term stability - will make them extremely popular. The initial manufacturing probiems with bubble memories have been overcome. According to Venture Development Corporation, bubble memories will be a $226 million dollar business by 1985.[10]

Because bubble memory is priced higher than disk systems (including floppy disk systems) and because it has poorer access times compared to main memory, the initial applications of bubble memory have been in the areas where moving disk memories are not suitable, like factory floors, or where resistance to shock is required, for example, in automobiles and portable terminals. As the cost of bubble memory decreases, it will surely be used as a substitute for semiconductor and disk memory systems.

## Hardware Strategies

What does all this mean? First, it is imperative that directors of university computing facilities maintain a very flexible position with respect to installed equipment. Equipment should be leased, and leases should be short term. Purchase of equipment should be avoided whenever possible.

---

[10] See *Computer Business News*, 22 September 1980, p. 21.

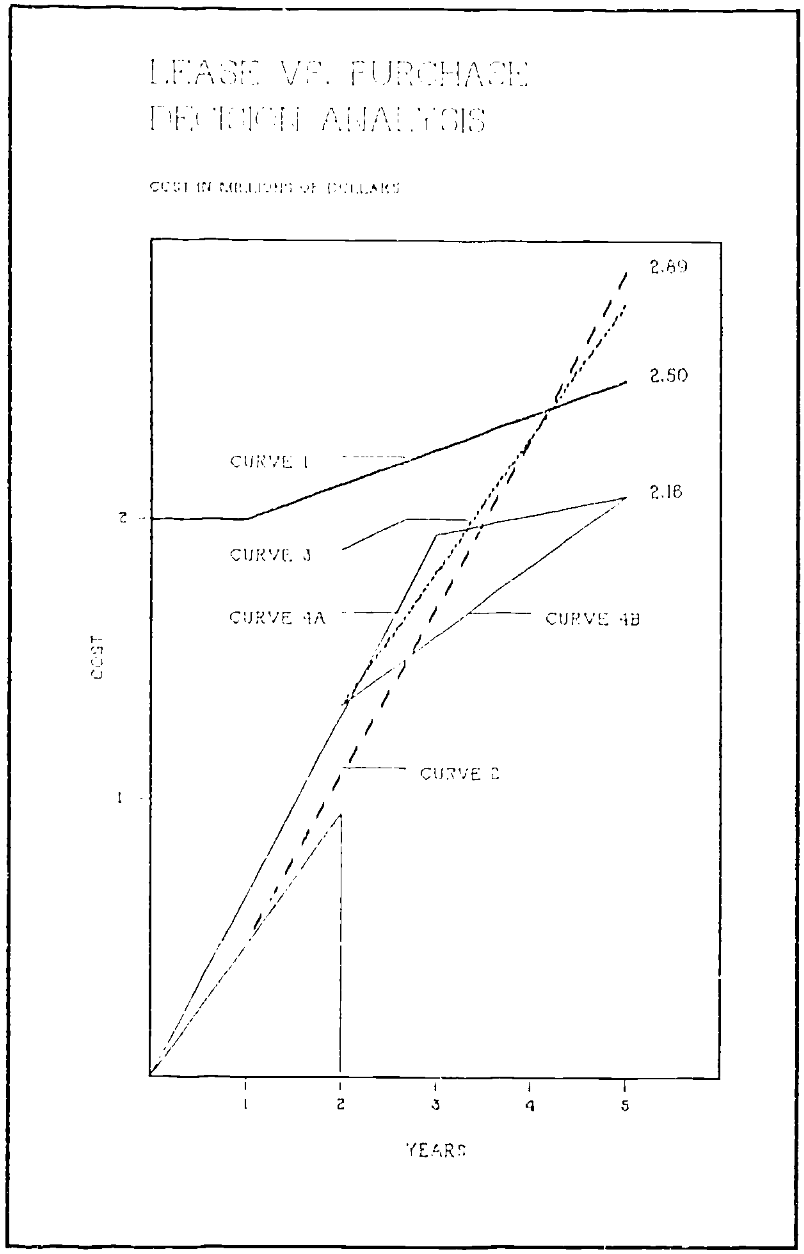

LEASE VS. PURCHASE
DECISION ANALYSIS
COST IN MILLIONS OF DOLLARS
2.89
2.50
2.16
CURVE 1
CURVE 3
CURVE 4A
CURVE 4B
CURVE 2
COST
2
1
1
2
3
4
5
YEARS

Figure 18

An example of an evaluation procedure to evaluate the lease versus purchase decision is presented in Figure 18. Curve 1 shows the cost of outright purchase. The purchase price of the equipment is a little over $2 million. If purchased, the equipment comes with a year's warranty and hence there are no maintenance costs during the first year. The planning horizon is taken to be five years for a total cash outlay of $2.502 million. Curve 2 shows a five-year lease purchase agreement using a third-party lease. Curve 3 shows a straight lease cost over a five-year period. Notice that the lease cost includes maintenance. The shaded area 3A corresponds to the accruals which are shown for a two-year period. If we make the very conservative assumption that prices are going to drop by 30% over a two-year period, then the purchase price of the equipment at that time will be approximately $1.4 million. The accruals are $936,000 during that period, making the net purchase price less than half a million. If this sum is financed over the next three years, we will see a reduction in cash flow shown in curve 4b. On the other hand, if we maintain the same cash flow the machine becomes fully paid for at the end of the third year. Years four and five simply have the maintenance costs, as shown in curve 4a. This analysis was done for an actual procurement during the summer of 1979. The recommendation, based on the analysis, was to lease, with the strategy that if within two years the price dropped at least 30% to go ahead and buy the machine unless new evidence dictated a different strategy.

Second, the useful life of computer equipment should never be taken to be more than five years, with four years more reasonable, and three years perhaps more accurate. Depreciation allowances, where permitted, should be based on 36- or 48-month schedules. Replacement allowances should also be based on similar schedules. If local, state or federal regulations prohibit these schedules, efforts should be directed at changing these regulations.

Third, charging schemes for computer usage that are based on a usage charge of 6 2/3% per year should be abandoned and replaced by more suitable and realistic figures. A 6 2/3% rate inherently assumes a 15-year equipment life. No computer is expected to have such a life. Even though computers are theoretically built to last forever, factors like technological obsolescence, increased maintenance and operational costs preclude a long useful life. For instance, the power requirements of an IBM 370/158 are five times as much (22.8 KVA) as the IBM 4341 (4.4 KVA). It also generates five times the amount of heat

(64,900 BTU compared to 13,800 BTU). The total difference in cost of electricity at today's prices can be around $12,000 a year.

Fourth, we have already said that only a part of the improvement in price performance will come from improved technology; the rest will come from economics of scale. Thus, unless there is overriding consideration to do otherwise, those products that have — or have a potential of having — a large installed base should be chosen.

Fifth, the dramatic price decreases in the industry have all but negated Grosch's Law which states that twice the price buys four times the computer power. Currently, for equivalent computer power, smaller computers are no more expensive than larger computers. Is it better to buy a large number of smaller computers, or a small number of larger computers? The answer to this queston is not so straightforward or one sided as it used to be.

In favor of single or small number of large computers are arguments, like increased efficiency of machine and personnel, larger capacity, and shared software. Amongst the disadvantages of larger systems are greater overhead, larger incremental expansion and greater impact of equipment failures. The overriding factors today are related to personnel and software. As will be seen later, the cost of computing will be governed not by hardware costs but rather by personnel, software and communication costs. Whereas there is no clear-cut recommendation for a general case, it can be shown that minicomputers are extremely cost effective if they can be dedicated to single application areas. Examples in administrative processing might be library automation, food service management or medical records. Notice that none of these applications requires any degree of data sharing with other institutional applications like payroll, student records, etc.

Sixth, as disk memory systems become cheaper the function of tape systems for storing user data will be greatly reduced. Tape systems will be used primarily for backup and archival purposes. All operational systems will run from disk. Therefore, existing user jobs that heavily depend on tape drives should be re-examined with a view to moving them to disk systems. The argument against a tape system for anything other than backup and archival is two-fold: it requires operator intervention thereby slowing down system operation; and, most of all, it requires additional personnel resources to manage the tape library, to verify the quality and readibility of the tape and to assign and allocate tapes to users. The operating system resides in virtual memory; the user program resides in virtual memory; perhaps the time is soon coming when we should be planning for user data to reside in virtual memory.

## Personnel Problems

So far we have discussed the drop in hardware prices and its impact on the management of computing. The other half of Figure 14 deals with software and personnel costs. These costs are projected to increase at an annual rate of 8%. Let us examine the personnel situation first.

A report entitled, *Science and Engineering Education for the 1980's and Beyond*, released by the Office of Science and Technology on October 14, 1980 paints a very bleak picture on the availability of computer professionals. The shortages, the report predicts, will continue through the 1990s.

The very obvious solution to the problem is to train more computer professionals in our institutions of higher learning and therein lies the biggest problem. There are 67 Ph.D.-granting computer science departments in the U.S. and Canada. According to the Snowbird report, in 1980 there were 1300 jobs advertised for only 200 graduates with Ph.D. degrees. To make matters worse for higher education, of the 200 Ph.D. graduates, 100 were lured away to industry with potentially better research facilities and certainly better salaries. This has resulted in a very acute shortage of teachers and researchers to teach computer science.

There is another significant and alarming trend. In 1975 there were 256 Ph.D. graduates in computer science. By 1980 this number had dropped to 200. It is extremely difficult for computer science departments to attract students to the Masters and Ph.D. programs. Also according to the Snowbird Report, bachelor degree holders get starting industrial salaries averaging $20,000, Masters $26,000 and Ph.D.'s $32,000.[11] With the very tempting salaries for B.S. degree holders it is increasingly difficult to attract students to the graduate programs. The lack of graduate students ultimately is reducing the number of individuals

---------------

[11] Peter J. Denning, ed., "The Snowbird report: A Discipline in Crisis," *Communications of the ACM*, June 1981, pp. 370-374.

capable of teaching computer science, thereby contributing to the inability to train more undergraduates to meet market demands.

It appears that our institutions of higher education are not only unable to produce the requisite number of graduates to meet the demands, but there is also some question of whether the graduates who do enter the job market have the proper background and skills. The mission of the computer science departments, according to Peter J. Denning (president of ACM and head of the Computer Science Department at Purdue Univesity), is to educate computer scientists, not to train students for their first job as computer programmers in business data processing organizations.[12]

In examining statistics published by IBM we find that before 1970 only 16% of the employees working in the information system area came with a computer science background; the remaining 84% came from business, finance, engineering, math and other disciplines (with over 25% from math).[13] Since 1970, 44% of the employees came from computer science and only 56% came from other disciplines (only 8% came from math). These statistics, though not representative of the industry as a whole, do point to a significant trend. The field of information systems is becoming so complex that it is increasingly important to hire individuals with specialized training. The pool of graduates we choose from is therefore smaller than it used to be, making this yet another factor that contributes to the shortage of professionals.

This acute shortage of personnel is projected to continue for several years at least through the mid-eighties. The shortage is already pushing salaries higher. Studies show that salaries are expected to increase between 20 to 23% from 1979/80 to 1981/82.[14] This leaves no doubt that

---

[12] Philip J. Gill, "University Shortcomings called factor in Personnel Shortage," *Information Systems News*, 26 January 1981.

[13] William R. Bradshaw, Jr., "IBM Examines its Personnel Planning," *Information Systems News*, 26 January 1981, p 45.

[14] Dunhill Personnel System, "Career Outlook '81," *Information Systems News*, 26 Jauary 1981, p. 40.

the salary portion of the curve in Figure 11 will increase by at least 8% per year, perhaps more.

### Software

Software development continues to be a very labor intensive process. Advances in software development techniques have fallen far behind advances in hardware technology. Programmer productivity, in spite of all the advertised tools available in the marketplace has not improved substantially. Industry estimates indicate that programmer productivity has improved an average of 3% a year. Software maintenance also continues to be a very time consuming and expensive process. Thus software costs are very closely linked to personnel costs. There is a notable exception: we find that application software for microcomputers is relatively inexpensive. This is true because of the large numbers of potential sales of software for microcomputers and because a large portion of the microcomputer software is developed by hobbyists in their spare time. It is a cottage industry.

Since software costs are linked very closely to personnel costs, it is expected that its rate of increase will follow the trends for personnel costs, at least until more advanced software development tools become available.

### Impact of Software and Personnel

What then will be the impact of increasing software and personnel costs on the management of computing in higher education? What strategies should be developed to cope with the changing situation?

First, and perhaps foremost, lies the need to make the salaries available to computer professionals in higher education more competitive. This problem is particularly acute in state supported institutions. Several states are re-examining their compensation programs with a view to bringing them more in line with the realities of the business world. Failure to do this will simply make our institutions the training grounds for those able to offer more attractive compensation packages. The salary issue has yet another dimension. The salary commanded by a systems analyst may easily exceed that of a tenured associate professor or, say, the director of personnel. Data processing directors command even higher salaries -- sometimes as high as the president of a small institution. These higher salaries can cause serious discontent amongst the ranks of other professionals in the institution. Salary structures in some cases are unable to deal with these differences. These problems have been partly responsible for some institutions contracting for the management of their computer facilities. The contractor receives a consulting fee and hires the necessary individuals to provide the services. The institution, therefore, does not have to carry these disproportionately higher paid individuals on its payroll.

Second, we must re-examine the jobs (not the job description) of our computer professionals to determine what portion of their present assignments does not require any computer-related talents. These unrelated assignments/tasks should be reassigned to other individuals. Basically the ratio of computer professionals to other staff in the computing center will, of necessity, decline. Plans should be made to operate with this changed mix.

Third, training and retention programs should be revamped. Institutions can offer special educatonal opportunities to its employees at marginal costs. Keeping an employee challenged and trained reduces the possibility of turnover.

Fourth, efforts should be made to reduce or eliminate labor intensive tasks. Labor intensive tasks should be discouraged by first providing alternate services and then pricing the labor intensive services artificially high. Thus keypunching should be discouraged by providing development terminals to programmers, by the design of better software systems, by capturing data online, and through the use of optical scan forms.[15] Because of operator intervention requirements, the use of tapes for application programs should be discouraged. Similarly, mounting special forms, bursting multipart forms and in fact the use of the printer itself should be [illegible].

Fifth, it is becoming clear that hardware vendors will make all efforts to reduce their operating system software support costs and also protect their investment in that software. Thus there will be a continuing trend for vendors *not* to locate software support staff at user sites but rather provide the services through a centralized response

---

[15] One might be tempted to argue that "keypunching" is not a productive use of the programmer's time but actual experience indicates that online programming is indeed more productive.

center. Further, new releases of software will come with greater frequency than before. Both these factors will make it more difficult and more expensive to maintain user modifications (local mods) to vendor supplied operating system software. Hence a systematic plan must be developed to review all local mods with a view of eliminating them. The user modifications slated for elimination should be staged for removal from the system with the installation of each new release so as to reduce the user impact. In any case, as more and more operating system functions get built into the hardware or microcode, the luxury of making local mods may cease to exist.

Finally, an environment should be created in which users can do more and more of their own computer-related work. This would apply both to the operation of information systems and to the development of software.

In summary, the economics of computing promises to change and change significantly. A larger and larger portion of total costs of computing will come not from hardware but from software and personnel. Energies are therefore better expended in arresting the growth of these latter costs. Lower hardware costs will necessarily result in the proliferation of computers. This, in itself, is not bad, for computers are not known to pollute the environment, use any significant amounts of scarce resources (the world has an abundant supply of silicon) or cause any exotic ailments. On the other hand they do hold a promise - a promise of greater productivity and lower costs - both essential ingredients in restoring faith in our economy.

# ORGANIZING FOR COMPUTING

"...some minds are stronger and apter
to mark the differences of things,
others to mark their resembiances."

Francis Bacon

## Combined vs. Separate

Ever since we discovered we could use the same computer for keeping student records and teaching high energy physics we have been locked on the dilemma of whether we should attempt to do both tasks on the same machine. The proponents of combined computer operations point to Grosch's Law to support their point of view on the basis of economies of scale. The proponents of separate operations point to the difference in machine resource requirements for compute bound and input/output bound tasks to support the position that a machine that does one task well must, perforce, do a poor job at the other. The argument is exacerbated by security issues. The impact of data integrity, security and privacy on administrative operations is keenly felt and generally talked about if not actually monitored. While security issues are every bit as important in the academic community, too few faculty recognize just how significant an impact a security breach (or lax security provisions in general) could have on their instructional or research efforts.

Both points of view are overstated and new technology, both software and hardware, is constantly changing the parameters of the argument. The general acceptance of sensor-based, process control has broadened the argument between those who see the machine as a data processing device, input output bound and programmed in COBOL; those who see it as a computational device, compute cycle bound and programmed in FORTRAN; and those who view the machine as an automatic control device, instruction set bound and programmed in assembler. A new generation of computer users weaned on graphics, interactive languages such as APL or very high level, interpreted programming systems probably sees it as all three. Even given the overstatement of each point of view, the dilemma remains. Networking, once viewed as the solution, now clearly has only moved the domain of the problem from hardware to software.

The issue of combined vs. separate has historically been resolved in favor of combined, until such time as a "critical mass" of computing capacity has been developed. At that point, the prospect of migrating to a new system with different control language, access methods, language nuances, etc., creates an inertia against change. On the other hand, encouraging change is the prospect of more powerful compilers, special purpose peripherals, or whatever. If a 30-year history is any indication, most organizations can look forward to a continuing debate, first resolved in favor of one concept, and then the other.

The first three generations of computers brought us from slow, hands-on, small storage, limited peripheral device systems to fast, interactive, mass storage systems. In the process of this transition the computer system began to be seen in depersonalized, assembly line terms. This transition predictably left the user feeling disenfranchised and estranged from the whole process of computing. The mystique of the "computerman" developed and was accompanied by an increasing anxiety on the part of those people who depended on the results of data processing but were not computer literate. With increasing numbers of business people trained in computing as part of their university education we can expect to see an amplif..ation of the pressure for using computers as we use the telephone - an increase in the support for distributed (networked) computational capacity.

Coming hand-in-hand with the question of combined vs. separate is the issue of where to settle in the centralized/decentralized spectrum. This is not a hardware issue but rather a question of how most effectively to develop and utilize software. At this stage of computer technology this is probably the more important question. Many of the same issues economy of scale, transferability of data, duplication of peripherals, etc. impact the decision on centralization. But of overriding importance are management issues. While it is useful to speak of the effect on the bottom line on the corporate balance sheet, in fact, the centralization issue is seldom ever translatable to such simple, unambiguous terms (and it is not clear that many universities have a bottom line expressed in dollar terms). To what extent to decentralize is more typically a question of management style and philosophy. In administratively decentralized organizations one would expect to see a complimentary decentralized software development structure and just the opposite for highly centralized organizations.

There is not an optimal level of software development centralization for all organizations, but rather an appropriate level for any given organization which best suits its organizational and managerial structure. Top management personalities aside (which is a BIG aside), the university may be viewed in several perspectives. Functionally it may be a teaching learning campus selling a product education. Geographically, it may be an enterprise of diversified activities operating over a wide geographic region. It may also be seen as a high technology, research and development enterprise with supporting manufacturing and information dissemination arms. And so on, for a number of other perceptions of the university, all held by different people at the same time. These viewpoints might lead to any number of configurations for software development and program support.

In many universities, data processing has ended up in accounting or the controller's office because of the historical accident that the first data processing efforts began there. Whatever the ultimate choice, for whatever reasons, it is increasingly likely that there will be some university data processing staff with at least oversight responsibility. This seems not only unavoidable, but justified, in the light of computing expenditures which commonly amount to five percent or more of the operating budget.

## Management Techniques

Too many data processing systems are counted as failures. This statement applies equally to the hardware complex as well as specific software development projects.

Failure is measured in many ways, budget overruns, schedule slippage, poor response time, inadequate system access, excessive hardware down time, and so on. The root manifestation is simply user dissatisfaction with the product delivered. Today's network environment only exacerbates the perception of failure. If it is frustrating to cope with problems when you can talk to a knowledgeable operator, imagine how much more so when you are at the other end of a 300 mile telephone line and all you can get is a busy signal.

Terminology may well be at the root of many, if not most of these problems. We have long since graduated from processing data to a cybernetic world of information systems. It is a dangerous anachronism to refer to what takes place in the university computing complex as "data processing." The processing of data in the historical context of the phrase makes up a very small subset of the larger problem of managing information systems. To many users, the computer complex is totally transparent (even invisible) and is seen only as a high speed printer, or only as a text editor, or only as an archive, or only as a cash register, or only as a switching device, or only as ... you probably get the point. Management must learn to see the computer network in the same light as it is viewed by a myriad of disparate users — as an information system, not just something which processes data. We feel so strongly about this point that we shall hereafter refer to the totality of hardware, software and people as an information system and attempt to deal with the question of information systems management.

One of the frustrations of managing information systems is the user's seeming intolerance for, and lack of understanding of hardware problems. The fact that 98% of the system is operating perfectly hardly placates those users who need to use the 2% that is down. If one is a believer in Murphy's Laws, then the 2% which is down will effect at least 40% of the population. Any well designed system should provide some level of hardware redundancy commensurate with the probability of failure and mean time to repair of hardware components. This is particularly critical in a teleprocessing environment where a dozen five second failures are significantly more disastrous than a single 20-minute outage. The terminal user who sees the system go down twice a week during his connection will understandably view the entire system a failure.

While the number of hardware systems perceived as flops is a well kept secret, software failures are a constant subject of coffee table gossip. Software failures are certainly easier to identify and more pervasive in their visible manifestations. One reason for this may be our propensity to plan, specify and set performance standards in such copious detail for software systems. Their failure to meet criteria are obvious. In many cases, performance criteria for the hardware complex have never been established and failure is judged subjectively rather than objectively.

Before discussing formal organizational structure it is only fair to reflect on the structure that actually dominates the organization. We have all heard that it is really the sergeants who run the Army and the secretaries who run the office, and there is more than a little truth to both observations.

The best analogies to the business organization are those drawn from biology. The organization is like a living organism. Some of the cells are dying and being replenished, other functions are atrophying from lack of exercise, some cells are out of control and must eventually be excised for the good of the whole organism. So it is with business organizations.

Irrespective of a person's slot in the organization chart, his or her influence and capabilities are seldom discernible by measuring the number of levels removed from the top of the chart. Personalities, politics and persuasion all contribute to getting the project approved, funded and properly completed. The discerning manager finds ways to use all three, in concert with the formal organizational structure, to accomplish the task at hand.

One of the hallmarks of the successful information systems manager is the ability to identify the underlying informal organization and to bring it to bear on the problem at hand. It seems that large organizations are destined to become complex and bureaucratic. The capability of utilizing the informal organization is sometimes referred to as "cutting red tape" or "bypassing the bureaucracy." In any case, the larger the organization the more critical it becomes to understand the informal organization and to be able to mobilize it.

Most contemporary management structures are hierarchical, that is the reporting structure is a pyramid. Complementary, compatible or coextensive functions are grouped together in so far as the top of any sub-portion of the pyramid is able to coordinate and administer the activities. This coordination is generally referred to as span of control which is typically measured in terms of the number of subordinates a supervisor can control. Optimally this is three to six, with eight or ten representing the upper limit of effective control. This process is repeated vertically until a single top point is reached, hence the pyramid.

Hierarchical structures occur frequently in nature and anthropologists postulate that it is the natural organizational structure for man. People join in families, families in clans, clans in nations, etc. The prevalence of hierarchical management structures has led to the terminology of top, middle and functional management, where we distinguish top management by its global, long range viewpoint; functional management by its local, short term view; and middle management as information filters and facilitators passing information back and forth between the other two, providing a feedback control channel. The operations (console operators, tape librarian, printer operators, etc.) section of a Computer Center is an area within the Information Systems Department where hierarchical organization has worked well.

The difficulty with hierarchical organizations is the lack of clear, organization-wide goals and the fragmentary nature of information possessed by its employees, particularly those at lower levels in the pyramid. This is one of the factors contributing to the lack of understanding of information systems in the organization at large.

One of the hallmarks of software development has been its inability to deliver a successful product under normal hierarchical management structures. Most major software development projects currently feature some form of development team. Employees may be members of several concurrently active teams, leading to a management structure in which the organization chart is more like a table or matrix than it is like a pyramid.

The structure probably developed as a consequence of the early, extreme specialization of people in the computer field. Scientific and commercial programmers quickly parted company over the issues of programming language and storage requirements. The control software programmers are quickly parted from them both. The rise of data base management systems, teleprocessing systems, networks of teleprocessing hardware, etc. have all contributed to fragmentation through specialization.

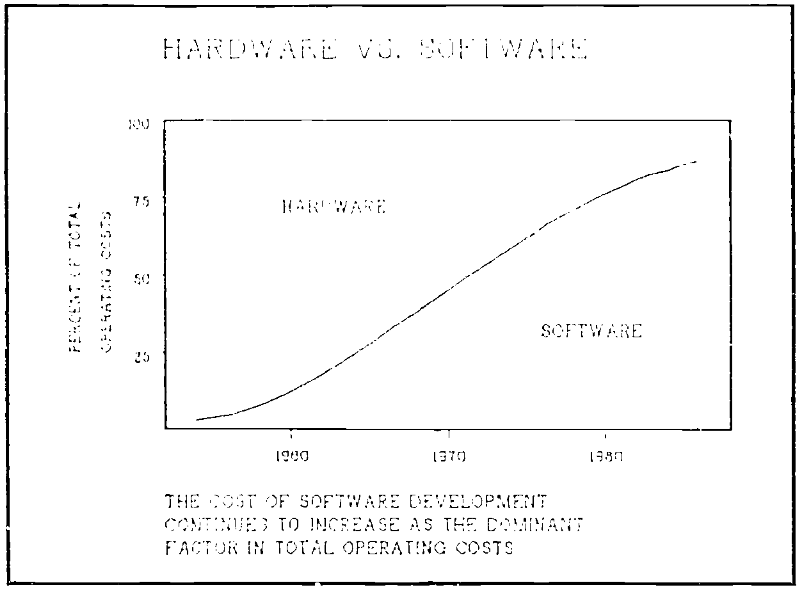



Figure 19

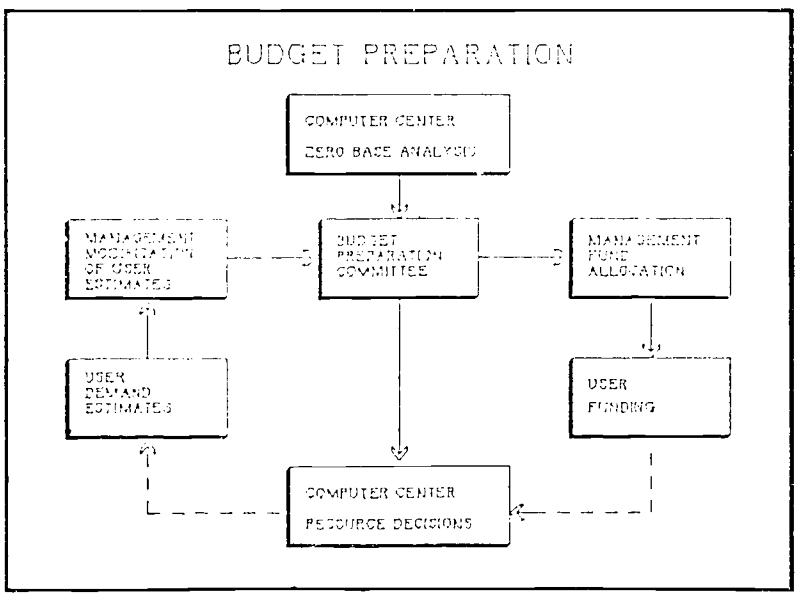



Figure 20

It is increasingly difficult to imagine any major software (or for that matter, hardware) development effort that does not require the coordinated efforts of specialists drawn from [illegible] of these areas. Frequently these skills are needed only on a part time basis, but over a substantial part of the development cycle. Hierarchical management structures are not well adapted to handle such temporary, part time arrangements. The team concept, a variant of matrix management, is the prevalent industry response to managing development projects.

## Reporting Structures

There seem to be two basic methods for determining the reporting structure of the Information Systems Department. One is to find some amenable vice president who is willing ([illegible]) to put up with the headaches, the other to find the organizationally rational reporting point for Information Systems. Having said this, one observes a lot of the former in the latter. Certainly Information Systems contributions to the total organization can be stifled by a boss who is neither knowledgeable nor sympathetic. Most of us quickly perceive the advantages of an active, articulate boss who is willing to take the time to understand the problems and carry potential solutions on up the managerial ladder. Equally apparent are the disadvantages of reporting to a boss caught in a backwater of the university who is unable to bring to bear the concerns of his peers for Information Systems problems, no matter how sympathetic he may be to Information Systems related problems.

A quick look at to whom and where the computing enterprise reports will say a lot about how much Information Systems can, or should attempt to, accomplish. While Information Systems are the backbone of many universities, there are as many more where its role is less fundamental. Ambitious projects require strong management backing, something which may not be possible to procure in situations where Information Systems has a weak reporting structure or is not perceived in the mainstream of the university enterprise.

The information processing function can generally be identified as existing within one of three typical organization structures. Most large organizations are already overburdened with staff reporting directly to the chief executive officers. Newly established data processing operations are frequently placed in the staff role, however. As the operation grows and matures it is generally spun off to one or more of the line operating divisions. There is a trend to revive this relationship for the organizational Information Systems Coordinator (and perhaps a small staff) with programming, systems analysis, and operations reporting to established line managers. In those organizations and industries where Information Systems is extremely important, there is an increasing tendency toward the establishment of a Vice Presidency for Information Systems or some such title. The more typical environment is one in which the Information Systems organization exists at the department level. Extreme managerial liaison and understanding may be necessary to pursue a service center role in this environment.

One of the contributing sources of lack of good communication between Information Systems and users is the structure of the Information Systems Deparment. A glance at the organization chart of the Information Systems Department will generally be sufficient to predict the status of Information Systems/user relationships. Surprisingly, many Information Systems Departments have no formal user contact point anywhere in their structure. This situation seems roughly equivalent to a computer vendor having design, engineering and manufacturing divisions but nary a trace of a sales force. (Come to think of it, we believe we have dealt with a few like that.)

The ultimate, if not usually immediate, consumer of Information Systems services is located external to the Information Systems Department. Some provision for formal contact must be made in the structure of the Information Systems Department, the higher in the structure the better. In general, we could argue that a "sales" or "consumer affairs" position should exist at the level immediately below top Information Systems management. As this position becomes further removed from top management, the user correctly perceives his influence degraded and senses that his opinions and needs count for little.

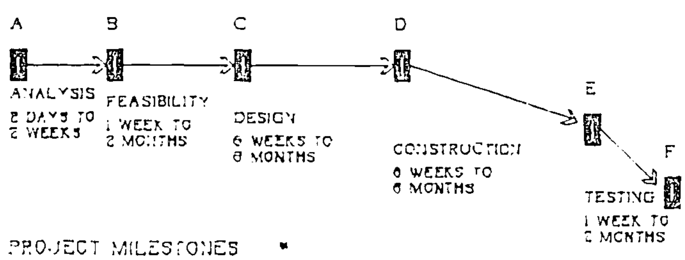
One of the real sadnesses of many computer organizations is their almost total introspection. They operate as though in a vacuum, totally div[illegible]ed from the needs, aspirations and expectations of the rest of the university. This estrangement frequently surfaces when machine utilization records are reviewed. The introspective computer department frequently accounts for 50% of all computer resource utilization. While this may be evidence of poor management in many cases, it is always evidence of an internally focused operating philosophy.

## SECURITY AND DISASTER PLANS

- **IDENTIFY POTENTIAL SOURCES**
  - A - RISING CREEKS AND FLOODING OF MACHINE ROOM
  - B - APPLICATION PROGRAMMERS CAUSING SYSTEM CRASH
  - C - OPERATOR MOUNTING WRONG FORMS
- **EVALUATE DOWN-TIME COSTS AND CONSEQ. DAMAGES**
  - A - 4,000,000 - IN DOLLARS
  - B - 1,000 - IN DOLLARS
  - C - 300 - IN DOLLARS
- **ESTIMATE PROBABILITY FREQUENCY OF OCCURRENCE**
  - A - 0.000000001
  - B - 0.001
  - C - 0.0005
- **COMPUTE RELATIVE COST OF PROTECTION**
  - A - 4,000,000 * 0.000000001 = 0.004 - IN DOLLARS
  - B - 1,000 * 0.001 = 1.000 - IN DOLLARS
  - C - 300 * 0.0005 = 0.150 - IN DOLLARS

Figure 21

## TYPICAL PLANNING HORIZON




PROJECT MILESTONES *

- A - UNABRIDGED OPTIMISM
- B - UNCRITICAL ACCEPTANCE
- C - WILD [illegible]
- D - TOTAL [illegible]
- E - SEARC[illegible] GUILTY
- F - PUNISH[illegible] OF THE INNOCENT AND PROMOTION OF THE NONPARTICIPANTS

* WITH APOLOGIES TO R.C. CANNING

Figure 22

This kind of an operational philosophy acts as a brake on innovative computer applications and fails to exploit the ubiquity of the computer. Good user relations are fostered in the Information Systems organization that sees itself as a purveyor of resources and a facilitator or catalyst in the man machine synergism. The rejoinder to such an open philosophy is that we should soon "have the inmates running the institution." The stampede in some organizations to the personal micro mini is little more than the user voting with his feet, expressing his feelings regarding poor service from the Information Systems Department.

## Job Shop vs. Utility Operation

Widespread acceptance of networking will likely render moot the issue of job shop vs. cost center operations. Our initial mode of computer usage was probably most akin to that of a library. The emergence of the microcomputer will probably engender a resurgence of demand for this hands on use of computers. However, where economics of scale have indicated the need for a large scale system, the debate over whether to operate it for hardware or people efficiency will continue. The manufacturing industry has rather conclusively demonstrated that where the machine is the critical resource, job shop scheduling techniques can produce significant economies in production. For those of us in computing the critical question revolves around the assumption that the machine (in our case, the computer hardware) is the critical resource. All recent industry statistics show quite clearly that software and software development costs are increasing significantly, while hardware costs as a percentage of total operating costs are decreasing. In fact, many industry observers suggest that software costs already exceed hardware costs and should shortly represent something on the order of 75% of total operating costs. If this be the case, job shop operation with scheduled run times, pre-specified resource requirements and predetermined priorities make less and less of an impact on cost reduction efforts.

To significantly reduce the costs (or improve efficiency of the total operation), we need to first attack those areas where the expenditures are largest software development. This suggests that rather than optimizing hardware we should be optimizing the efforts of the people who write the software. Such a point of view permits a less optimal use of the hardware resource if that consequently permits improved efficiency in software development. Such a mode of operation is commonly termed a cost center priorities determined by purchasing power, computing resources sold to the highest bidder so to speak. We will speak more on this issue later, but we should be careful not to confuse the cost center philosophy with the idea of buying hardware to obviate management problems. Cost center operation requires as much, if not more, management control than does job shop operation.

A simplistic approach to viewing the problem might be to construct a decision matrix in which mode of operation is measured against computer system requirements. Such a matrix could consider the importance (weight) of the requirement to overall computing resources as well as which mode of operation contributes most to the satisfaction of any particular system requirement. The results of such an analysis will invariably favor the cost center or utility approach.

Perhaps the most difficult problem faced by Information Systems management is preparation and defense of budgets. This frequently happens because top management sees Information Systems to be somehow outside the normal management structure. Certainly for computer intensive organizations it is hardly ever possible to sell support levels at anywhere near the expectations of Information Systems users. This frustration frequently leaves Information Systems management in the precarious position of viewing their users as the "enemy." Quite the opposite should be the case.

This pent-up demand on the part of the users should be the strongest weapon in the budgetary arsenal of the Information Systems manager. Those operations run as cost centers should additionally be able to harness user competition for priority of service as an additional selling point for expanded Information Systems operations.

To ensure equitable Information Systems budgets, it is probably appropriate for the Information Systems manager to see that computing needs are viewed in the context of total university operations at budget time. Being a part of the usual push and shove of budget making has many advantages, but perhaps the greatest is the latent demand of users which can be used to create a strong case for better budgetary consideration. Information Systems management should exercise some restraint in dealing with the issue sure to rise: "Why can't we spend our money on outside services, or for paper clips?". Unless the computer operation can bank profit (not normally recommended in most intra-university operations), it has no way to adjust adapt to the marketplace dynamics and planning uncertainties when its normally "captive" customers are given complete freedom in regard to spending their Information Systems budgets. However, consistent with the general management philosophy of the university, computing dollars should be exchangeable for other dollars and vice versa.

The options for providing Information Systems services to the university are many and varied. It is becoming increasingly common to purchase both specific purpose software and hardware services. The "not invented here" syndrome seems finally to have fallen prey to the realities of economics. Some of our very best software packages are marketed by companies not in the mainframe business. Special purpose software is frequently contracted for from consulting firms which specialize in specific segments of the Information Systems marketplace. Less typical is the use of general purpose service bureaus in lieu of in-house data centers, or facilities management contracts for the operation of the in-house center. However, the national time sharing networks still represent one of the fastest growing segments of the Information Systems marketplace. In fact, many universities got their first experience in interactive or online systems with the time sharing networks.

It seems increasingly likely that the cost conscious Information Systems manager will look to the speciality suppliers for hardware and software services that he lacks expertise or critical mass of users to support. It is curious to observe the number of Information Systems managers who still offer no interactive services to their users on the basis that they lack the hardware or control software to support online systems. Many such managers will see their centers lost to a plethora of micros and minis if they do not find some way to satisfy the user's increasingly voracious appetite for online systems.

The conventional batch Information Systems mode of operation was introduced with the first generation machines in the mid 1950s. To complement this mode of operation, batch processing languages such as COBOL and FORTRAN were developed and well established by 1960. It rapidly became apparent that the full potential of the computer would not begin to be realized until indivi[illegible]sers were given personal access to the system with [illegible]atural, or at least more interactive, programming la[illegible]. By 1970 the teletypewriter and CRT had become reliable I/O devices used in a time sharing mode on most larger systems. About the same time, interactive languages such as JOSS, BASIC and APL were developed to facilitate the man-machine synergism.

During this same time period, significant advances in the use of the computer as a "real time" monitor and control device were made. Since then, exceptional advances have been made in improving the one real weakness of the computer — its input/output capabilities. Plotters, wand readers, computer output microforms, digitizers, mark readers, remote sensors and even voice recognition equipment are on the market which are reliable and cost effective. Unfortunately, particularly among the smaller Information Systems operations, we still see the philosophy and techniques of the 1960s being applied to the problems of the 1980s.

## Security and Disaster Planning

Most Information Systems Departments lack a specifically designated, long range planning function. This is most frequentlv attributable to the small size of the Department, but frequently is the case because everyone assumes someone else is doing the planning. While we could not overemphasize the importance of such a designated position in the organization, we must nevertheless recognize that such a position probably does not exist.

Information Systems software exhibits one of the extreme forms of technological obsulescence. Most major software efforts require from two to six years to complete and their time to technological obsolescence is about the same. While this does not seem to have inhibited many software development projects it certainly creates severe problems in the planning area. Even if software development efforts are amortized over a five-year period (which may be good bookkeeping practice) we are hardly ever ready to begin major system re-design in so short a period of time.

In fact, many good software projects will attempt to anticipate the state of the art several years hence so as to prolong the useful (or economical) life of the project. Even so, we are dealing with hundreds of thousands, if not millions, of dollars of development effort. The optimum payoff to the universitv will not occur if these software developments are not married to the enterprise's long range plans. One could not emphasize too strongly the desirability (if not necessity) of having the Information Systems staff represented at university planning sessions.

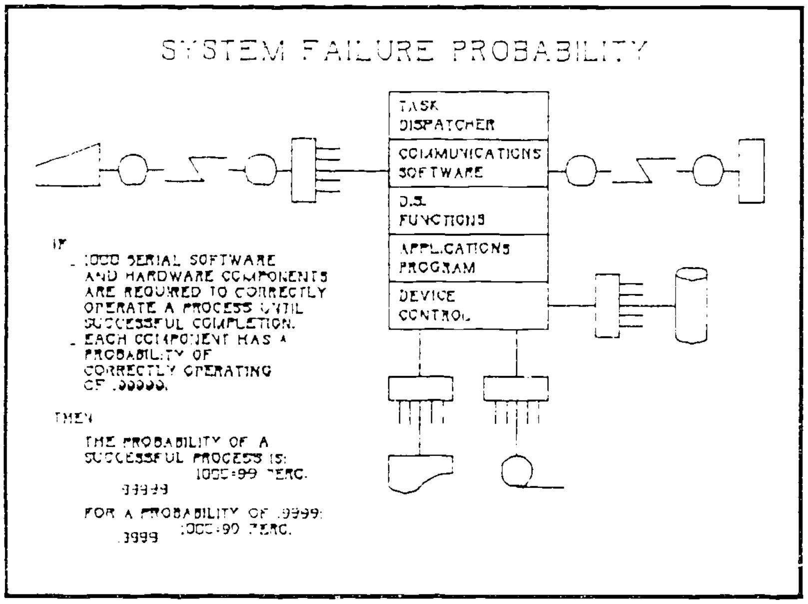



Figure 23

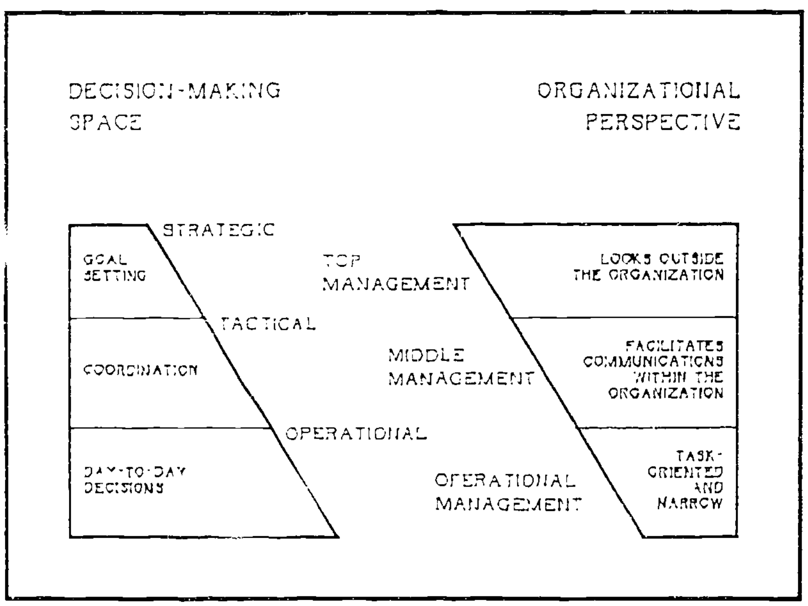



Figure 24

At one end of the scale such representation provides the Information Systems staff the opportunity to react to immovable lead times before they have become frozen in the university development plan. At the other end, the positive contributions to the educational enterprise from Information Systems can be folded into the long range plans and develop that top management awareness and empathy so necessary at budget time.

One of the planning problems which typically is not addressed has to do with disasters. The definition of disaster depends on who is impacted and may range from a head crash on one of the disk drives to the total destruction of the hardware complex by fire. Either end of the disaster spectrum carries with it the potential of driving the university out of business, or at least creating an unbearable dislocation in its scheduled operations.

Part and parcel of planning responses to disasters is the recognition that fatal problems do not require catastrophic loss of hardware. The probability of serious loss by dissatisfied personnel bent upon revenge, poor physical security, and improperly supervised and planned systems or operations changes is much higher. Consequently, disaster planning should attempt to assess not only the magnitude of the loss (and the cost of consequent recovery) but the probability that such a loss might occur. One of the ironies of our profession is the Information Systems operation that backs up its critical files in at least three impregnable vaults or caves and has backup provisions to use any one of three other installation's hardware in the event of a catastrophe while allowing unknown visitors to roam the machine room and implementing no software security levels on a system which can be accessed by anyone who happens to have access to the public phone system.

We should also recognize that exposure may be equally as fatal as destruction. Security and privacy are concerns that go hand in hand. It is still considerably easier to access sensitive information in some user's trash can than it is to tap the data links or "trojan horse" your way into the computer system.

## Development vs. Maintenance

One of the most frequently heard complaints of Information Systems Departments is that they can not get anything done because they are always "fighting fires." One could observe that the fires probably represent the work of [illegible] within the Information Systems Department itself. A [illegible] of factors comprise the "fire fighting" syndrome. [illegible] understanding (often abetted by the Information Systems Department) between the user and Information Systems leaves the user convinced that any change he [illegible] in the output format or computational envelope of [illegible] is [illegible] for the asking. Inadequate system [illegible] contributes to the problem because the user's needs have been viewed as static rather than dynamic. Unrealistic [illegible] upon the computer operation, from data entry to [illegible] has left the process a mystery to the user and a lack of education also contributes to the problem.

[illegible] leads us to an infinite regress. [illegible] we are busy fighting fires we do not have time to [illegible] better understanding, which leads to more fires, which leaves insufficient time for good system design, which leads to [illegible] and so on, ad nauseum. An organization that finds itself lapsing into, or already deeply mired in, the "fire fighting" syndrome has a long term project if it is ever to recover. The "fire fighting" syndrome can be precipitated by a number of reasons: hardware reconfigurations, migrations, or upgrades; sudden and unexpected shifts in the volume of the work load; conversion from batch to interactive; hardware instabilities; and just plain poor systems design.

Surprisingly, one of the principal reasons for "fire fighting" is the development of systems that are intentionally designed to be supervised and run by computer professionals. The user Department interface with the system is intentionally limited to a formatted sheet for hand entered input data and some conglomerate output sheet that yields every conceivable status of the system. This design philosophy is frequently fostered by the "let us do it for you" attitude — an attitude with some rationale in the 1950s when Information Systems was trying to prove itself, but totally inexplicable for the 1980s.

For these and many other reasons, the computer operation becomes totally immersed in maintenence (and operation with its incident run books, etc.) to the almost total exclusion of development. Such obsolete, patched-up systems eventually breed a seige mentality on the part of the Information Systems staff. One non-solution frequently proffered is that "we will undertake no new development until we get our current shop in good order." Referring to our biological analogy, a system which is not discarding its old and malfunctioning cells and concomitantly generating new, flexible replacements, will soon perish.

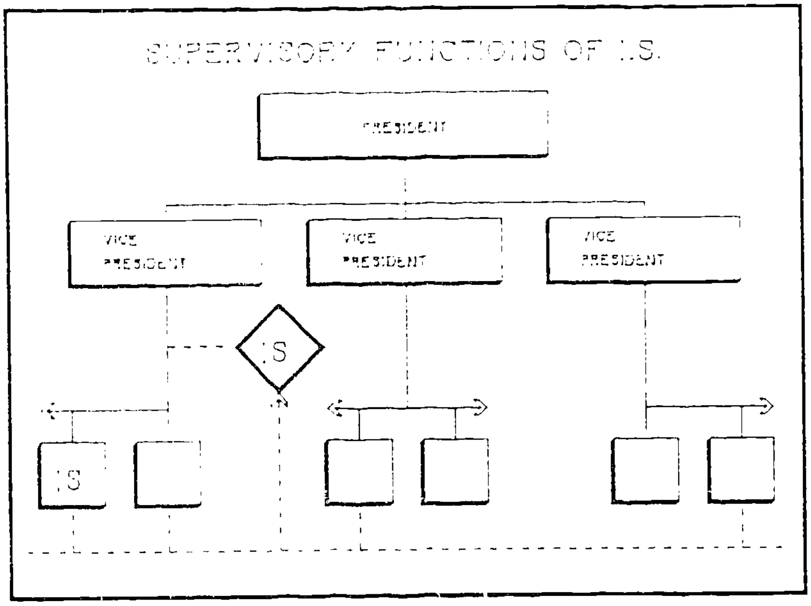



Figure 25

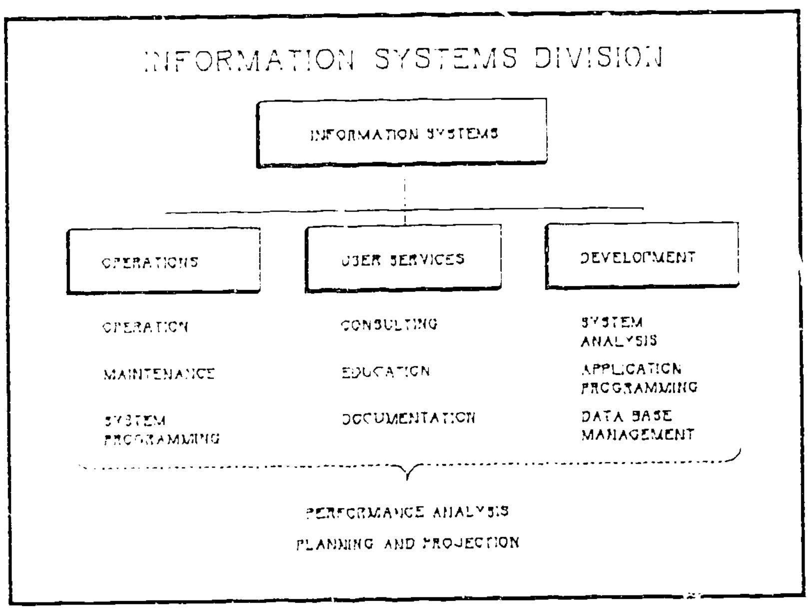



Figure 26

Most Information Systems Departments suffer from a poorly defined interface with their customers. What interface does exist is usually passive, reacting to user complaints and doing little to open formal lines of communication. The argument can safely be made, in the strongest terms, that the computer operation exists only to provide support services for the rest of the organization. Rather than telling other departments in the organization what they should be doing by computer, the Information Systems Department should be educating them in the capabilities of modern computer systems.

Every Information Systems Department should have a User Services Division, reporting at a high level, which sets the tone and tenor for development with the Information Systems Department. Education and consulting should be the hallmarks of this group. Perhaps the best model we could find would be the time sharing companies. In-house seminars, conducted on a cyclic schedule, should provide one of the most effective means of opening communication channels. Each seminars should be short, an hour or two a day, perhaps every other day for a week. The topic selected should suggest itself immediately in terms of the status and goals of the target departments. The choice of instructors is critical and should not necessarily fall to the person most knowledgeable in the particular topic. The "classroom" should be carefully chosen with an eye for informality so as to encourage open discussion and frank questions in a friendly atmosphere.

The User Services Division should also maintain a consulting station (and phone number) where knowledgeable members of the Information Systems Department spend their full time, prepared to answer questions, provide consultation on system problems experienced, and generally demonstrate the Department's desire to be as helpful as possible to its customers. Again, the most knowledgeable people are not necessarily the best consultants: tact and courtesy may be even more important than raw knowledge. A good consultant or consulting group will frequently identify problems before they become disasters. They should be in an excellent position to suggest modifications to the operating system or procedure libraries based on their continuing contact with user problems.

Summary information on system performance ought to be made available to users as well as top management. This might take the form of a well conceived annual report or some simple monthly or bimonthly newsletter. In either case, "news stories" or successful applications may do a lot to spark interest in computing in those dark corners of the organization that have been holding up information system developments.

## Evaluation and Audit

It is critical that the Information Systems Department participate in the process of evaluating its performance, even more so in those situations where top management will not or does not feel competent to. An outside evaluation should be conducted at least once every three years and more often during periods of hardware upgrades or extensive software development efforts. The outside evaluation can be one of the most potent weapons the information systems manager can wield at budget time as well as helping considerably in improving understanding on the part of top management and major user departments.

An appropriate evaluator or evaluation team could be selected from a list provided by the Information Systems Department to top management. Several of the big eight accounting firms have divisions that do this on a regular basis, and there are any number of consultants or firms which can do an outstanding job. If competitive pressures permit, the information systems managers of universities of the same general characteristics, who are responsible for similar scale operations constitute an excellent source. The cost of a reasonably comprehensive outside evaluation is modest when compared to benefits it will likely produce.

The audit should review university goals, performance standards and systems, and operational procedures. The auditors should be encouraged to identify operational strengths as well as weaknesses. Such an evaluation should also concern itself with budget, reporting structure and other management concerns. We are always amazed at how the outsider seems able to catch the attention of the boss and get him involved in a problem which we have been telling him about for years. A good audit will generally investigate security and backup provisions, compensation levels and staff and user satisfaction. While one never expects to escape unscathed from this kind of searching inquiry, a well managed operation will be perceived as just that — and it may be a plus to identify areas that need more management attention.

As well as encouraging external review of Information Systems operations a strong case can be made for continuing internal oversight. Unlike external audit, this function is not performance review, but rather policy review. Such internal groups are frequently called steering committees, thereby designating the goal establishment rather than performance evaluation task.

Such a committee should be relatively small, six to eight members, and chaired by the person to whom the Information Systems manager reports. Its membership should be drawn from management on the same or higher level as the Information Systems Manager. Its frequency of meeting will also be highly correlated with the level of new developments in the computer operation. The chairman needs to insure that the committee does not descend to the day to day operational level, but rather concerns itself with long range planning, strategy and performance standards. The manager of the Information Systems Department should use the group much as many boards of directors are used — to [illegible] ideas for future expansion, to delineate priorities and establish the performance standards expected of the computer operation.

Such a committee can become a strong force in opening communication channels, both to the user community and to other elements of top management. Discussions in this committee should both broaden the Information Systems Manager's perception of the whole organization, as well as significantly improve other managers' perceptions of the importance and contribution of Information Systems to the total business enterprise. The committee could consider such items as the impact of a new teleprocessing system — its potential for negative impact on existing systems — and could become a strong voice for the necessity of hardware upgrades to avoid degradation of current systems. The committee could help define performance standards (average or minimal acceptable response times, etc.) and become a strong lobbyist with top management for budgetary changes to help achieve these standards.

Kenneth Kolence, developer of the PPE and CUE software monitors, has characterized the field of computer performance measurement as "a set of numbers in desperate search of unifying principles."[16] We find ourselves not only confounded by how to measure but also, unfortunately, what to measure. Even given some intuitive sense of what

---

[16] Kenneth Kolence, "Software Physics and Computer Performance Measurement," *Proceedings of the ACM 25th Natonal Conference*, August 1972.

[illegible] how to measure, it is still unclear how the measurements are to be interpreted!

This problem is not restricted to the hardware or control software components of the system, but extends to the performance of computer operators, programmers and systems analysts. The interaction of all the components makes it frightfully difficult to decide how to evaluate the performance of any specific component. Is operator mishandling due to poor coding, which is in turn a direct product of inadequate system design? What constitutes good coding, lines of code per day, fast execution, sparing use of storage, etc.? And how, if at all, are any of these related to the bottom line of the university balance sheet?

We know that the establishment of criteria will implicitly affect the strategy and goals of our personnel. If we choose to measure lines of code produced per week, we will not be surprised to see that the output of our programmers (in terms of lines of code per week) begins to increase. At the same time, we will not be surprised to see our run time and software maintenance problems begin to increase commensurately. One could make similar analogies in the hardware or system control software aspects of the computer operation. All this suggests that we should exercise more than a little care in making the basic decision on measurement criteria. We need to avoid measuring "things" because they are easy to measure every bit as much as we need to avoid *not* measuring "things" because they are difficult to measure.

While measurement may in itself be a science, the use to which we intend to put the measurements is hardly so objective. Invariably, we will engage in many subjective evaluations of worth, or what the economist would call utility. Criteria need to be established in a framework of something we wish to optimize and this infers subjective value judgements. We continuously need to differentiate between "doing the thing right" and "doing the right thing."

Most larger computer systems have some interface to the operating system to permit capture of data at the dispatching level of the system. IBM's SMF is typical of what is provided by the vendor. In general, these interfaces provide the capability to access raw counts of basic software operations, but provide little or no analysis. Since the hardware itself is normally not monitored, conclusions about the behavior of the hardware need to be drawn inferentially.

The time and effort required to build a performance analysis system are anything but insignificant. For a computer system of any substantial size and complexity there ought to be some such system. This clearly suggests that there should be some individual or group specifically charged with gathering, analyzing and distributing performance information. Because of the necessity for interfacing with the operating system this group might logically be drawn from the systems programmers who have responsibility for maintenance of the operating system.

Performance monitors fall in two classes, those that keep track of the hardware and those that keep track of software. Each type has its own particular strengths and weaknesses. As most data processing personnel and their managers came up through the software and programming ranks, they quite naturally think in terms of software monitors. With the increasing proliferation of teleprocessing terminals and complex communication data processing networks, we can expect to see more use of the hardware monitor.

Perhaps the best way to tackle the question of what to measure is to establish the results that are desired. Obviously, desirable results will include minimum response time for teleprocessing applications, well defined and minimum turn around times for batch applications, minimum cost for units of effort such as accounts receivable, invoices, etc. On the hardware side we might include a minimum total up time, a maximum length for a single down time, an upper limit on busy signals, etc.

These kinds of macro standards should be readily understandable by anyone with a nodding acquaintance of computer usage. They might well represent "guarantees" to the user community which are supported by distributed summary statistics on some regular basis. Many, if not all, of them will be explicit or implicit in the design of the various systems or applications. Such standards should not be treated as goals but rather as minimum acceptable performance levels. Do not expect any applause for bettering them, but be sure that a failure to meet them will focus a large amount of unfriendly criticism on you. Remember that if three components, each with a 90% probability of being available are required for the successful completion of a task, there will be less than a 75% (.9x.9x.9) chance of completing the task.

Given some reasonable set of macro standards one can then begin to focus on more specific, internal goals. Whether these should be postulated as goals or standards will necessarily depend on their interaction with other uncontrollable variables. Among hardware related measurements we might include excessive I/O errors on tape and disk drives, excessive errors on transmission lines, arm/channel interference, degraded printer performance, etc. On the software side we might track excessive program paging, task and program abends, device allocation delays, data set overflow activity, etc.

All these micro measurements will be established to provide information for performance improvement which contributes to meeting the macro performance standards. We need to be careful not to elevate the desire of minimizing something such as arm contention on the disk drives to the level of a performance standard. It is, after all, a problem in sub-optimization that has a rather unclear impact on the global optimization problem. We need to see the total system—hardware, software, and people—in much the same terms as a hi-fi system, i.e., only as strong as its weakest component!

Service charging in the job shop mode is probably best handled by unit charges on the user's view of the output—checks written, invoices printed, ledger items posted, etc. If such a charging methodology were rigorously adhered to we surmise that much of the internecine warfare between the Information Systems Department and its customers could be avoided. It needs to be clear to both parties in an implied contract just what the conditions of the contract are. The Information Systems Department cannot guarantee or be held responsible for the accuracy of output reports. They can, and should be, held responsible for the accurate and timely processing of data. Late submissions cannot be expected to be corrected by the Information Systems Department. Poor management practices in user departments cannot be corrected by good (even excellent) management of the Information Systems Department any more than the reverse. The dismal history of the data processing job shop leaves us wondering if it is a mode of operation that will ever work smoothly in any but the smallest of computer centers.

# SOFTWARE DEVELOPMENT

"Man's mind stretched by a new idea never goes back to its original dimensions."

Oliver Wendell Holmes

## The Data Base Approach

Data base concepts are relatively new. They were first introduced in the late sixties. It was not until the mid-seventies that data bases started being widely used. Today hardly anybody would consider the development of a management information system without the use of the data base approach. Since information systems so heavily depend upon the data base approach we will begin by defining some fundamental concepts about data and data bases. This information is presented primarily as background for the discussion toward the end of this chapter.

A data base is nothing more than a collection of data that can be accessed by a set of computer programs. This is a very basic definition of a data base which we will refine somewhat later in this chapter. The smallest identifiable unit of data is called a *data item*. A data item is also called a *field* or a *data element* however the term data item is the one that is becoming more widely accepted. A data item might be an employee name or student grade. Generally data items are grouped together to form a *record*. It might, for instance, be logical to group together student number, student name, course number, course credits and grade to form a record to be used in the grade reporting system. A named group of data items within the record is called a *segment*. In the above ex[illegible] may wish to define course number, course credi[illegible] to consist of a segment called grade segment [illegible] grade segment is a part of the student reco[illegible] The [illegible]inition of a record and a segment will be used late[illegible] explain the data model and submodel.

Data base management is a systematic approach to storing, updating and retrieving data contained in a data base. The data base approach consists of two co-existing parts: an approach to the development of information systems and the use of a special set of tools in the development process. The data base approach emphasizes the following concepts:

- *Integrated systems*: The data base approach insists that integrated systems should be built for maximum efficiency. Thus we should not have an accounting system that is unable to share data and controls with the payroll, personnel and purchasing systems. Integrated system implies that each system has full knowledge of the existence of the other systems. Systems are not built in isolation. Systems that do not share data, like cafeteria management and student records, need not be integrated.

- *Data redundancy be controlled and reduced*: This means that, whenever possible, a data item is stored only once. For instance, if employee name is kept in the personnel system, it should not be repeated in the payroll system. Each time the payroll system needs the employee name it fetches it from the personnel system. This method of non-redundant storage avoids inconsistencies in the data base. Corrections are easier as they have to be made in a single location only and automatically become available to all users.

- *The data base be treated as a resource*: The handling and treatment of the data in a data base should be similar to other important resources like space and money. Further the data base should be treated as a shared resource. The individual assigned to manage the resource is called a *Data Base Administrator*. Treating the data as a resource emphasizes the point that the data captured within any given system (like the payroll system) has value to the organization that goes far beyond the confines of that particular system.

- [illegible] a data base administrator is assigned [illegible] of the data [illegible] it is possible to [illegible] standards. These standards may [illegible] with the coding of data, [illegible] that [illegible] the data, with [illegible] of data.

- [illegible] the data base administrator has complete [illegible] of the data resource, proper [illegible] and [illegible] procedures can be developed [illegible]. This is a very important [illegible] without proper security [illegible] in the ability of [illegible] their interests.

- [illegible] points listed above are very [illegible] data base approach. However, [illegible] the data base approach is [illegible] What [illegible] by data independence? [illegible] program is [illegible] in such a way that it [illegible] of the data that is stored, [illegible] on the storage device and of [illegible] device then it [illegible] new data item is [illegible] data is changed or the [illegible] the program has to be [illegible] that no changes are [illegible] by additions of new [illegible] of data [illegible] in the meth- [illegible] of the increasing [illegible] discussed in Chapter 2, this [illegible] is crucial in containing the [illegible] and [illegible] costs of software systems.

[illegible] The data base approach [illegible] integrated systems in which the [illegible] Data Base Administrator (DBA). [illegible] advantages like non-redundancy of [illegible] of data, better standards, [illegible] and, most of all, data indepen- [illegible] but implied in the above dis- [illegible] of a software system called a [illegible] system that helps accomplish the [illegible] the data base approach. The term [illegible] sometimes used as a synonym for Data [illegible] System (DBMS).

### Architecture of DBMS

A simplified version of the architecture of a data base management system is presented in Figure 27. The top of the figure shows the physical storage of data. The physical storage is device dependent. There is a one to one mapping between the physical storage and the conceptual model of data. The DBMS accomplishes this mapping. In a data base environment, no single system or program is permitted to see the entire data model. In fact each program sees only that portion of the data that it needs to use. The rest of the data does not even exist for the program. This restricted view is accomplished by the data submodel (see Figure 27). Using the DBMS, the program requests data based on its view of the data—its submodel. The DBMS retrieves the required data. It first translates the user request that is based on the submodel to an equivalent request based on the data model. Using the data model it then retrieves the requested data and places it in the work area for the programmer to use. The process is reversed in the case of updating. The program places the data in its loading dock—work area—and requests the DBMS to update the data base. The data is processed through the submodel to the model and finally to the storage location for updating.

The programmer (or user) sees no more than the work area and the data submodel. The DBA has a broader view which extends from the physical storage to the data submodel. The figure shows that the DBMS has seven tasks to handle. The numbers in the figure correspond with the numbers in the following list of tasks.

(1) Managing the physical storage of data.

(2) Accomplishing the mapping between physical storage and the conceptual model.

(3) Permitting the definition of the conceptual model.

(4) Accomplishing the mapping between the conceptual model and the submodel.

(5) Permitting the definition of the submodel.

(6) Transferring data between the submodel and the program work area.

(7) Recognizing the definition of the work area in the program.

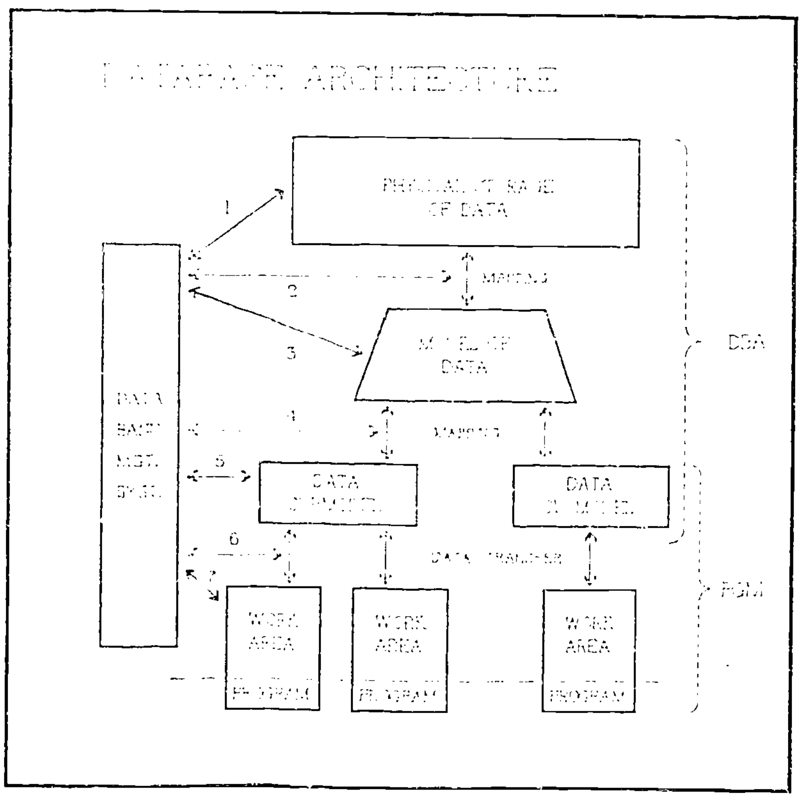

ARCHITECTURE
OF DATA
1
2
3
DATA
4
5
DATA
DATA
6
WORK AREA
PROGRAM
WORK AREA
PROGRAM
WORK AREA
PROGRAM

Figure 27

Data Base Management Systems also come with a comprehensive set of utilities. The most important of these utilities are the ones that allow for the backup and recovery process. These utilities ensure that in case of a system disaster the entire data base can be reconstructed. The reconstruction requires the use of the last backup and the use of the log records recorded since the last backup. Most Data Base Management Systems have a utility that monitors all the activity and keeps a record of the updates (and if desired, accesses) made to the data base. This process is called logging and the records created by it are called log records. In addition to the backup, recovery and logging utilities, the DBMS has utilities for performance monitoring, security definition and control and resource management.

## Models of DBMS

There are three conceptual models for data organization that have been proposed and are in use at this time. These models are hierarchic, network and relational. The hierarchic model arranges data in a tree like structure. The hierarchic model is straightforward. Conceptually, each record in a hierarchic data organization looks like a corporate organization chart. It is a tree structure with the root of the tree at the very top. The record itself consists of segments that are arranged like people in an organization chart. There is a rather rigid parent/child relationship amongst the various segments. This organization is very restrictive in that each segment can have only one superior. In some hierarchic organizations the rigid parent/child relationships amongst segments can be supplemented (but not circumvented) by techniques like logical relationships and secondary indexing. These techniques allow multiple access paths to a single segment at the cost of increased complexity. The network model is more general in that a record can have more than one immediate superior. The relational model comes closest to the natural way of storing data in the form of tables.[17] These tables are called "relations." The rows in the tables are called "tuples" and the columns are called "attributes." All possible values in a column form the "domain" for that attribute. An important characteristic of the relational model is that

---

[17] E. F. Codd, "A relational model for large shared data banks," *Communications of the ACM*, 6 June 1970, pp. 377-387.

[illegible] between tuples are represented solely by data [illegible] drawn from a common domain."

[illegible] and the network models are the most [illegible] models today. The relational model is the easiest [illegible] understand and the hardest to implement. Till very recently, the relational models implemented were very inefficient for large data bases. Several companies have recently [illegible] DBMS based on the relational model. There is, however, insufficient data to judge if the implementations available today are suitable for very large systems.

The DBMS software either includes or has the ability to be used in conjunction with other related software. The three software packages most commonly used to supplement the capabilities of a DBMS are a query facility/report generator facility, terminal control software (this is required for specialized terminals) and a data dictionary. The query facility/report generator is used directly by the end user. The terminal control facility is a programming aid for the programming staff. The DBA is the major user of the data dictionary.

The Data Base Management System is certainly a powerful software tool in the hands of the developer. It provides enormous capabilities; its use is essential in the development of large integrated information systems. Several data base systems are available through computer vendors and software houses. Picking the right data base management system is an important responsibility of data processing management. It is a subject matter that would require a text of its own to discus.

## Data Administrator

The Data Base Administrator has to tell the DBMS what to do. The tasks include the definition of the content of data, the storage structure, the access method, the authorization procedure, the edit checks and the backup and recovery procedures. The DBA has several tools to accomplish these tasks. The foremost amongst these tools is the Data Dictionary. The data dictionary contains macro data or data about data. The dictionary allows for definitions of data items, cross-references of the use of data items, cross references of programs using the data and validation codes for data items. Other tools available to the DBA are data definition languages that define the model and submodels of data; mapping and display tools used in the design of data bases; simulation tools to test concepts in design; monitoring and tuning tools to ensure proper resource allocation and performance. It is clear that the task of a Data Base Administrator is a technical one. The DBA is a corporate technical manager of data.

There is a growing recognition that top management needs to be actively involved in the planning and development of information systems. If data are truly to be treated as a corporate resource then all data, both computerized and non computerized, should be managed by the top ranks in the administration. Determining who should have access to what data, determining the relative priorities of the information system development process, determining how various units of the university are tied together by the information links are all institutional decisions of some importance. The highest ranking administrator responsible for the data resource and precipating these decisions is sometimes given the title of "data administrator." Institutions of higher education prefer to use the title of Vice President of Information Systems or Associate Provost of Information Systems. In managing the information resource, we are identifying two organizational needs. A need to address the data base management responsibilities at the technical level and the need for information resource management at the Vice Presidential level. Organization structures to facilitate these dual needs are still in the stages of experimentation and evolution. Chapter 6 discusses this subject matter in some greater detail.

So far we have discussed the various components of a Data Base Management System. It, and the related software, form the *major tool* in the development of information systems. Next we will look at the Systems Development Life Cycle. The Systems Development Life Cycle and the related structured methods are the *techniques* used in creating an information systems. These techniques are discussed next.

## Systems Development Life Cycle

The systems development life cycle is normally defined as consisting of six stages:

1. Project Initiation
2. Functional Analysis
3. Systems Design
4. Construction
5. Installation
6. Maintenance and Enhancement

[illegible] have been written on the subject of the [illegible]. We will only briefly discuss some of the very important points [illegible] stages. It is imperative [illegible] in [illegible] stages of the life cycle. [illegible] the completion of each stage the user should [illegible] the decision of whether to [illegible] with the project or abandon it. Only when the [illegible] the go no-go decision can we insure that the [illegible] developed is for the user and not for the [illegible] of the EDP department or [illegible] systems ana[illegible]

[illegible] the user? One very simple answer is [illegible] who pays the bill. Which brings [illegible] the second point. It is very important that the [illegible] associated with the development project be esti[illegible] by the Information Systems Department and approved by the user. Further, the user should receive periodic [illegible] that show the actual expenditure associated with the project. At project initiation time the status of the present system should be outlined, the objective of the new system should be agreed upon; and, most importantly, an acceptance criteria for the system should be outlined and [illegible]. The acceptance criteria should include func[illegible] testing, integrity testing, user training and complete documentation.

Next comes the functional specifications of the system. These specifications must be written in user understood language. It should be the first place one should think of looking if there are questions regarding the system. This of course means that it must always be kept updated. The functional specification should not only indicate what the system will do, it must also indicate what the system *will not do*. Many a system has failed because the user simply assumed that the system would do something it was only natural for him to assume so. It is, therefore, imperative [illegible] limiting scope of the system be spelled out very [illegible] the development cycle.

In the design stages, the user interfaces to the system play an important role. Much thought must be given to exactly what the user will be required to do in order to make the system work. Also of great importance is finding out what the user will be required to know in order to make the system work. For instance, it is not uncommon to [illegible] codes for curriculums in which degrees are [illegible]. If the system requires that the Hegis code be put into the system (as opposed to curriculum abbreviation which the student put on the form), then the operator has [illegible] the begin code or memorize the entire list [illegible] user interface.

The other important function in the design stage is the design of the data base itself. This is perhaps the single most [illegible] aspect of the entire process. Whereas DBMS [illegible] a great degree of flexibility, there are certain changes that are very [illegible] to make once the system has been built. DATAPRO research corporation lists three categories of changes in degree of difficulty that the DBMS can handle. Most DBMS are able to handle changes in [illegible] changes in blocking factors, adding new fields [illegible] deleting a field in a record, adding a new record and reorganizing a data base for efficiency without having any impact on the application program. Some of the DBMS, though not all, can additionally handle changes in size and format of a data item, new relationships between existing records, new access methods, space [illegible] changes in inverted keys for access without [illegible] impact on the application programs. However [illegible] changes that most DBMS cannot handle without [illegible] impact on the application programs. Examples [illegible] moving a data item to a new record, combining two records, splitting a record and changing the access key of a record. This list is by no means complete. It is presented here to show the relative importance of the data base design process. It not only determines what [illegible] currently possible to do now, it also influences what [illegible] economically feasible to do in the future. It is for this reason that a data base system project should be undertaken only after sufficient thought and planning has gone into the overall process.

Due to the relative importance of the data base design, it is good practice to have the design reviewed by peers. This review process forces the designer to articulate the rationale for the design and allows the reviewers to suggest alternative, perhaps better approaches to the design process. Experience indicates that it is useful to allow the design to age for sometime. Usually the designer is able to think up something that was overlooked or is able to find a better design.

Construction of the design should be in parts. These parts should be candidates for partial implementation. An important aspect in the programming of the system is its error handling capability. Each time an error occurs in the system it is important that the program completely specify the following:

- Type of error
- Location in the program where error occurred
- Location in the data where the error occurred
- What action the programmer should take

These four steps can save untold hours during the operation of the system.

After the system has been successfully implemented, we get into the most difficult phase of the life cycle — system maintenance. Maintenance today seems to be a problem of large proportions. In many installations, 70% of the programming staff is dedicated to maintenance. National averages are not too much lower. There are two partial solutions to this problem. One is an organizational solution discussed in Chapter 6. The other is related to the system. Structured techniques discussed later improve the readability and maintainability of the programs. Simplicity of design, small understandable modules, data independence and [illegible] tables in program logic all help [illegible] more maintainable. Maintenance of existing programs is a necessary and expensive requirement. However, management commitment to new development should be such as not to take resources away from development in order to meet the growing demands for maintaining old programs. Enhanced capabilities and improved functionality will continue to be the right priorities through the 1980s. Resources to develop, buy or otherwise introduce these advanced capabilities should be explicitly identified and budgeted for.

## Structured Techniques

E. W. Dijkstra in March of 1968 expressed a view that goto statements were harmful. This was the beginning of a whole new methodology called structured techniques. Structured programming was the first of these techniques to come into the mainstream of the profession. The last three years has seen the rapid development and use of other structured techniques like structured analysis, structured design and structured walkthroughs.

Structured programming is a method of programming that follows the rules outlined below:

- Avoid goto statements

- Use only three basic structures
    - Simple sequence
    - Selection
    - Repetition
- Use only a single point for transfer into the structure and a single point of control out of the structure
- Use clarity of presentation
    - Indent to show the hierarchy of logic
    - Small size modules (50 lines)
- Use a top down design.

The objective of structured programming is to compose programs correctly rather than debug them into correctness. Figure 28 shows the relative error rates found in design, programming, testing and operations. It clearly indicates that system design is the most critical part of the development process. Design errors found in operations are by far the costliest. Information systems management should focus its attention to this area and should try to reduce the errors in this category.

Structured analysis is based on two key elements. The first element is the building of a graphical model (also called logical model) of the system to be built. The second element is a verbal description of the system objectives and constraints. The graphical model is an excellent way of showing the user exactly what the system is supposed to do. This along with the verbal description of constraints makes for a concise description of requirements. This model is then used to determine what portions of the system are to be left manual and what portions will have development priorities over others.

Structured design is a mechanism which permits the graphical diagram of the system to be converted into small modules. Each module should have the property of communicating with a minimum number of other modules. This is achieved by arranging the modules in a hierarchy such that each module communicates either with its immediate superior or its immediate subordinate and with no other module. This hierarchy can be achieved by studying the flow of data through the system. An example of hierarchic breakdown into modules is shown in Figure 29.

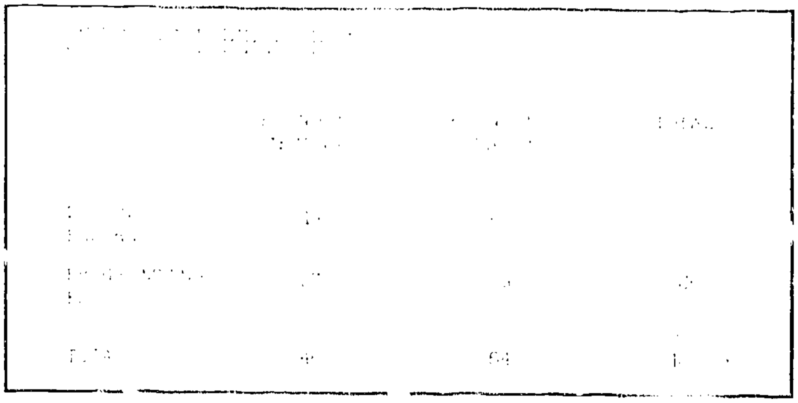


Figure 28

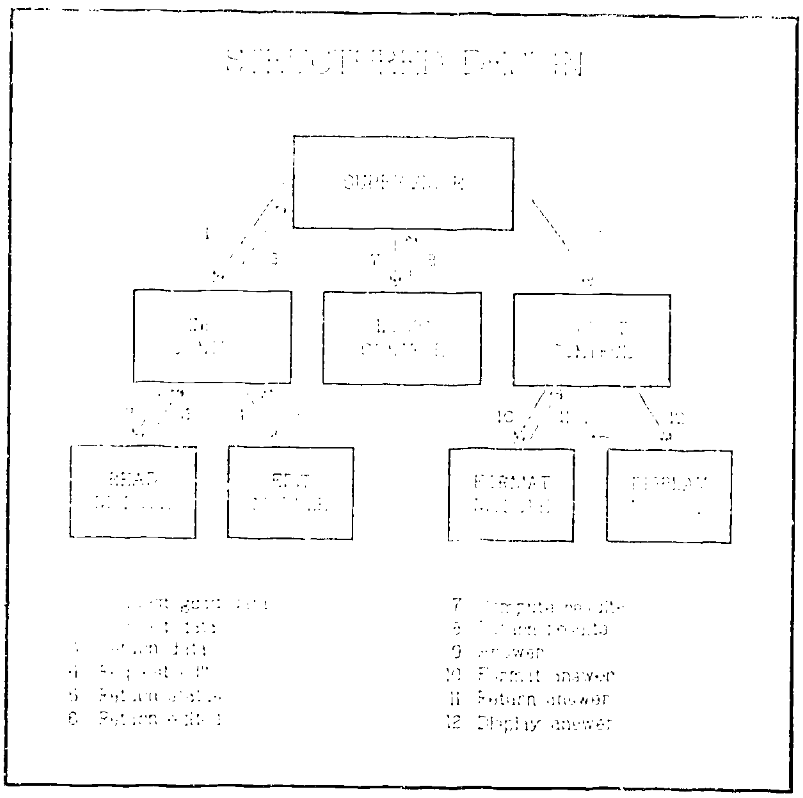



Figure 29

Structured walkthrough is no thing more than a systematic way of presenting the system design to your peers for their review.

Structured methodology is extremely good in that it tends to systematize the development process. It is, however, not the answer to the software problem for it produces efficiencies in the range of 12 to 20%. What we need is a methodology that improves the efficiency of software development by a factor of 10, or 100 or even 1000.

## Problem Areas

The technique for system development described in the previous pages suffers from several problems. It assumes that the user knows the needs and therefore is able to specify them. This is not true. Many users do not know what is required. This is particularly true of first time users. Information systems normally go through three stages of development. The first stage is a stage of substitution. Users perceive the computer being most useful for performing certain manual functions they are presently performing. The second stage is the stage of backfill. Once the computer is in an application area, more and more applications become obvious and the computer is used in a natural sense. The third stage is one in which not process, but the task itself, is analyzed to determine how best the computer fits in. Most organizations are in the backfill stage and the techniques described previously are all structured to analyze and automate existing processes. What we fail to recognize too often is that many of the existing processes were designed for an entirely different technology long before computers were available. Automation is great, but what we need is innovation with automation. Innovative applications will only be possible if we can tear ourselves away from the existing processes and examine the broad objectives that we need to achieve and then devise processes that more fully utilize the tremendous communications and processing capabilities of the computer.

The System Development Life Cycle as defined has too many steps. It takes entirely too long between the discussion of objectives and the delivery of the product. In light of the increasing personnel costs this large time frame will drive many system development groups out of business. The present development process has too many "communication" steps in it: writing programming specifications (analyst to programmer), writing functional specifications (analyst to user), writing design specifications (analyst to [illegible] and [illegible] [illegible] guides (analyst to user), [illegible] testing [illegible] (by analyst, programmer) and [illegible] manager). All of this [illegible] professionals who are in [illegible] stages. [illegible] will continue to be a very [illegible]

[illegible] are two [illegible] that might help the situation. [illegible] shows the user involvement in the development [illegible]. The user involvement is very heavy during the [illegible] stages, then drops off during the design and con[illegible] stages and picks up again during testing and [illegible] the project to be more effi[illegible] [illegible] involvement should be [illegible] and steady as shown in the lower half of the figure. [illegible] are [illegible] benefits to this. Because of contin[illegible] less formal communication is required. [illegible] user can [illegible] the user's guide and a [illegible]. This way we can guarantee that it will be written [illegible]. The [illegible] involvement will also help [illegible] design errors. Figure 25 shows the percentage [illegible] versus [illegible] errors in systems. [illegible] truction [illegible] will [illegible] design errors [illegible] d.

Programmer analyst communications [illegible] written specifications are becoming unaffordable [illegible] should either be direct involvement of the program[illegible] in the project team or the analyst should be able [illegible] the code. The distinction between programmers and analysts is often arbitrary or blurred except in the mo[illegible] rigid of environments. In environments that are large and complex, where direct programmer involvement in the project is not possible, the analyst should use the prototyping technique to demonstrate the system both to the user and the programmer.

On the lighter side one can have quite a different perspective of the stages of building a corporate data base. These stages have been characterized as:

- Uncritical acceptance
- Wild enthusiasm
- Dejected disillusionment
- Total confusion
- Search for the guilty
- Punishment of the innocent

- [illegible] of nonparticipants

## Management Information Systems and Beyond

Having [illegible] the tools and techniques that are available for use today, it is time to discuss what exactly constitutes a management information system (MIS). It is perhaps easiest to define a management information system by saying what it is not. Having a data base management system to manage the data does not mean that one has a management information system. However, using the data base approach, an effective MIS can be built if one does the right things.

In Chapter [illegible] we discussed the perspective and needs of the various levels of management. At the operations level the functions to be performed are rather well defined and are process oriented. The registration process, the admissions process, the grade reporting process, the accounting and purchasing flows are all process oriented. Bringing the power of the computer to help perform these processes better will result in an *automated data processing system* (ADPS). It is certainly not a management information system. However, one must be quick to point out that without the automation of these operational systems, it is not possible to build a management information system. The data that is used in management information systems is gathered in the operational systems. In a sense then, management information systems presuppose the existence of automated operational systems.

The next level of management, generally called middle management, has need for information that is not necessarily process related but generally process generated. Examples of this information are answers to questions like: How many female undergraduates do we have in the college of engineering? How do our actual expenditures compare with budget amounts for full time on-campus faculty salaries? And so on. These questions are answered by aggregating the detail data that are generated in the operational system. This type of information is used for very well defined and structured decisions. A system that provides this information to the right individual, at the right time, and can reasonably insure that it is accurate and unambiguous can be termed an *information system* or *management information system*.

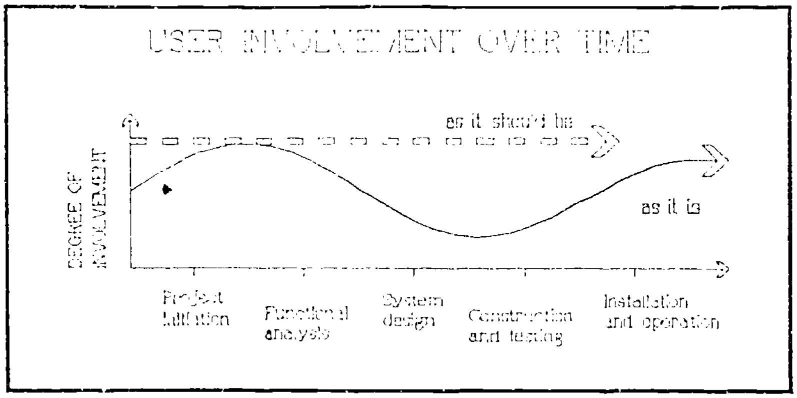



Figure 30

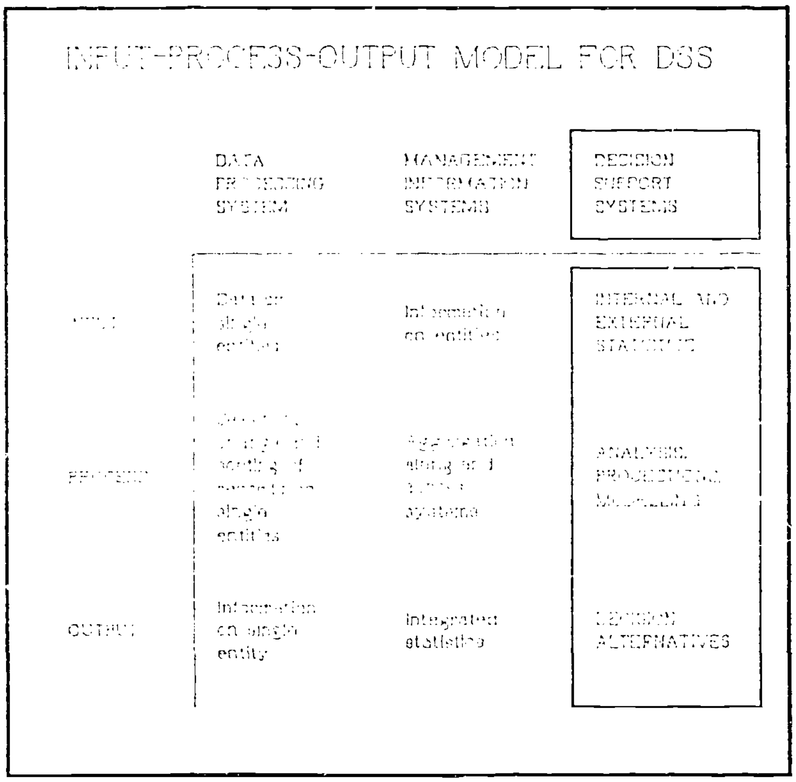



Figure 31

More recently, starting in the mid-seventies, a new term has evolved in the vocabulary of information systems. This term is *decision support system* (DSS).[18] Some consider decision support systems as merely another name for Management Information Systems. However, there is an implied difference between the two; the difference is primarily of emphasis and of scope rather than content. Decision support systems try to address the needs of higher management in making relatively unstructured decisions. In addition to the data generated in operational systems, typically they require the use of data external to the normal processes of the organization (like the state of the economy, the population growth factors, the cost of power) and the use of simulation, modeling and other analytical tools. When the analytical tools are built into the information system and can be used in a flexible and unstructured fashion we have a decision support system. Because of the complexity of the decisions to be made and the unpredictable requirements for external and internal data, systems with decision support capability are extremely difficult to build.

Any system can be defined in terms of input, process and output. Figure 31 shows the differences amongst a data processing system, a management information system and a decision support system using the input-process-output model.[19] A data processing system takes data on single entities as input and uses processes like *select*, *change*, *sort*, etc. to produce information on single entities (like grade reports) as output. The management information system begins where a DPS ends, taking as input information on entities. The process is usually aggregation along and across systems. The output of an MIS is integrated statistics (generally called "fighting" statistics). Decision support systems start with internal and external statistics as input and use processes like *modeling* and *forecasting* to produce decision alternatives as output.

---

[18] The September 1982 issue of CAUSE/EFFECT is dedicated to the topic of decision support systems. Along with several articles it contains a useful bibliography for those wishing to explore the subject matter in greater detail.

[19] The rest of this section is based on "Anatomy of Decision Support Systems," by V. Chachra and R.C. Heterick, *CAUSE/EFFECT*, September 1982.

A DPS is characterized by an operational perspective in which the information is prestructured and reporting is prescheduled. The decision process is generally rule driven and subject to prior optimization. An MIS on the other hand takes a tactical perspective and strives to provide on-demand, but predefined, information. Its use is more interactive than a DPS. A DSS tries to address the *ad hoc* information need for strategic planning. The requirements are unstructured and generally undefined ahead of time. The system therefore has to adopt to the user and allow for a subjective decision process. Figure 32 summarizes these characteristics.

It is not necessary that the only users of decision support systems be the top management of the instituiton. The integration of analytical tools with information systems makes a decision support system. Examples can be found in institutions where such systems exist for all levels of management and even for the benefit of students.

James Martin talks about yet another class of system, a system that is highly specialized, like an operations research system or a design system. It is not clear what characteristics make this type of system different from decision support systems. Perhaps it is the capability of advanced grahics integrated with the analytical tools imbedded in the information system that help create the operations research system.

How do institutions of higher education stack up in relation to this progression of systems? Most institutions have many of their normal processes (see FICHE list in Figure 33) automated. Several have rather sophisticated information systems. Modeling techniques have been in use on campuses for a long time. The most popular of these models were the student Flow Model and Resource Requirements Prediction Model, both developed at NCHEMS. More recently, there has been a growing interest in EDUCOM Financial Planning Model (EFPM). However, most implementations of these models have been "manual" in the sense that the data to support the model were laboriously entered into the model. In order to make effective use of models for institutional planning it is important that proper interfaces be built that allow institutional operational data to directly feed the models that use them.

## DSS CHARACTERISTICS

| | DATA PROCESSING SYSTEMS | MANAGEMENT INFORMATION SYSTEMS | DECISION SUPPORT SYSTEMS |
|---|---|---|---|
| ADAPTABILITY | User adapts to system | Half and half | SYSTEM ADAPTS TO USER |
| OBJECTIVE FUNCTION UTILITY | Predefined: subject to prior optimization or implementation | Identified but not predefined | DEFINED DURING DECISION PROCESS |
| DECISION PROCESS | Rule driven: objective | Interactive but objective | PERSONAL: SUBJECTIVE |
| PERSPECTIVE | Operational | Tactical | STRATEGIC |
| INFORMATION STRUCTURE | Prestructured | Structured | UNSTRUCTURED |
| TIMEFRAME | Prescheduled | On demand: predefined | AD HOC |

Figure 32

## PLANNING, MANAGEMENT AND INSTITUTIONAL RESEARCH

1-1 Budget Forecasting
1-2 Budget Preparation
1-3 Budget Analysis
1-4 Budget Position Control
1-5 Institutional Cost Studies
1-6 Faculty Salary Analysis
1-7 Support Staff Salary Analysis
1-8 Faculty Activity Analysis
1-9 Support Staff Salary Analysis
1-10 Induced Staff Activity Analysis
1-11 Resource Requirements Modeling
1-12 Student Flow Modeling
1-13 Long Range Planning
1-14 Enrollment Forecasting
1-15 HEGIS Reporting
1-16 Data Element Dictionary
1-17 Institutional Coding Structures

## FINANCIAL MANAGEMENT APPLICATIONS

2-1 General Fund Ledger
2-2 General Fund Expenditure Accounting
2-3 Departmental Expenditure Accounting
2-4 General Accounts Receivable
2-5 Student Accounts Receivble
2-6 Accounts Payable
2-7 Payroll
2-8 Employee Benefit Accounting
2-9 Retirement System Accounting
2-10 Bank Account Reconciliation
2-11 Cash Flow Projection
2-12 Investment Records
2-13 Investment Evaluation
2-14 Grant and Contract Administration
2-15 Research Project Accounting
2-16 Research Proposal Monitoring
2-17 Financial Aid Accounting
2-18 Tuition and Fee Accounting
2-19 Residence Hall Accounting
2-20 Stores Accounting
2-21 Telephone Accounting

Figure 33 - FICHE Application List

GENERAL ADMINISTRATIVE SERVICE APPLICATIONS

3- 1 Facilities Inventory (Space)
3- 2 Facilities Utilization Analysis (Space)
3- 3 Classroom Utilization Analysis
3- 4 Personnel Records
3- 5 Personnel Evaluation
3- 6 Personnel Placement
3- 7 HEW Compliance Reporting
3- 8 Staff Ethnic Group Reporting
3- 9 Civil Service Position Records

AUXILIARY SERVICE APPLICATIONS

4- 1 Faculty Staff Directory
4- 2 Faculty Club Billing
4- 3 Residence Hall Assignment
4- 4 Student Directory Preparation
4- 5 Student Housing Reports
4- 6 Food Service Menu Planning and Inventory
4- 7 Bookstore Inventory and Operations

LOGISTICS AND RELATED SERVICES

5- 1 Purchase Order Follow-up
5- 2 Purchasing Information System
5- 3 Vendor Information System
5- 4 Stores Inventory
5- 5 Office Machine Repair Control
5- 6 Equipment Inventory
5- 7 Automobile Registration
5- 8 Parking Lot Space Assignment
5- 9 Traffic Violation Records
5-10 Crime Reporting
5-11 Car Pool Matching
5-12 Motor Pool Records

Figure 33 (continued)

PHYSICAL PLANT OPERATIONS

6- 1 Physical Plant Accounting
6- 2 Physical Plant Job Scheduling
6- 3 Building Maintenance Costs
6- 4 Equipment Preventative Maintenance
6- 5 Key Inventory

ADMISSIONS AND RECORDS MANAGEMENT

7- 1 Undergraduate Admissions Processing
7- 2 Graduate Admissions Processing
7- 3 High School Testing Records
7- 4 Course Master Records (Catalog)
7- 5 Schedule of Classes Preparation
7- 6 Student Class Scheduling
7- 7 Tuition and Fee Assessment
7- 8 Student Registration Processing
7- 9 Class Rosters
7-10 Term Student Records and Reports
7-11 Course Add/Drop Processing
7-12 Enrollment Reporting
7-13 Enrollment Statistics
7-14 Student Ethnic Group Reporting
7-15 Term Grade Reporting
7-16 Honors Program Records
7-17 Student Transcript Master Records
7-18 Degree Requirements Evaluation
7-19 Correspondence Course Records

FINANCIAL AID ADMINISTRATION

8- 1 Financial Aid Evaluation
8- 2 Financial Awards
8- 3 Student Employment Records
8- 4 Work Study Records

Figure 33 (continued)

LIBRARY APPLICATIONS

9- 1 Acquisitions
9- 2 Cataloging
9- 3 Card and Material Preparation and Control
9- 4 Circulation Control
9- 5 Serials Holdings
9- 6 Bibliographical Search Service
9- 7 Fugitive (Ephemeral) Material Indexing
9- 8 Educational Media Services

OTHER ADMINISRATIVE APPLICATIONS

10- 1 Alumni Records
10- 2 Foundation and Gift Records
10- 3 Test Scoring and Analysis
10- 4 Curriculum Planning
10- 5 Teacher Evaluation
10- 6 Teacher Placement
10- 7 Fraternity/Sorority Rush Records
10- 8 Student Counseling Records
10- 9 Student Psychological Testing
10-10 Athletic Event Ticket System

HOSPITAL APPLICATIONS

11- 1 Patient Registration/Admissions
11- 2 Hospital Census
11- 3 Medical Records
11- 4 Appointments and Scheduling
11- 5 Central Supply Inventory
11- 6 Communications and Order Entry
11- 7 Dietary/Food Service
11- 8 Housekeeping
11- 9 Laboratory Information System
11-10 Radiology Information System
11-11 Pharmacy Information System
11-12 Nursing Station Support System
11-13 Physician Support System
11-14 Patient Billing/Accounts Receivable
11-15 Hospital Financial Information System

Figure 33 (continued)

## Some Design Guidelines

There are some very important design principles which if used in the design of information systems will result in a more effective operation and insure greater success for the system.

*Source point data capture.* Data should be captured in machine readable form at the point of its origin. A transaction reflecting a copying charge should be put into machine readable form in the copy center. A student wishing to process a drop or add should be able to do so at the departmental office perhaps in the presence of the course advisor. Accounting transactions should be in machine readable form at the department originating the transaction. This applies to pre-registration, wage cards, purchase orders or any other presently used input form. Notice that the objective not necessarily to place the data into the data base at the point of origin but at least to capture it in machine readable form. If it updates the data base – fine; if not, at least the data is in machine readable form. Source point data capture avoids duplication of work. It tries to eliminate the data translation process (punching cards from forms, entering into terminals from forms) and the errors inherent in the duplicate handling of the data. Combined with online updating, source point data capture becomes a very powerful mechanism for getting accurate and timely information into the computer system.

The idea expressed here is a simple one. If implemented, its impact on the design of the system and the operation of the institution will be far-reaching. It breaks down the traditional boundaries that now exist for the handling and processing of institutional data. It also changes the role of the administrative office pioneering this approach. These new roles are important and are discussed in Chapter 6.

*Value added data handling.* Data should flow directly from the source (origin) to the individual adding value to the data. The last individual adding value to the data should have the responsibility of updating the data base if the data base does not already contain the information. Thus a financial transaction that originates from a department might have value added to it by the accounting department that assigns an expenditure code to it. Notice the emphasis is on adding value, not control. Controls can be directly implemented in the data entry procedure particularly if it is an online procedure.

The flow of transactions should be examined. This can be done by examining document flows. All the "stops" in the flow should be identified. The question to ask at each stop is the following: assuming that the transaction is within prespecified guidelines (does not require exceptional handling), what value is added to the transaction (not control) at each stop? If no value is added then it is safe to assume that the transaction flows through the particular stop only for informational purposes. In such a case, the stop should be a good candidate for elimination in the forward flow of the transaction. Forward flow is the flow of information prior to completion of the transaction. Information requirements should be met by the reverse flow i.e., actions that are subsequent to transaction completion. Forward flow may be considered as the flow to the computer before and during the transaction processing life. Reverse flow is the flow of information from the computer after the transaction processing is complete.

*Destination point document generation*. This is perhaps the most important and counter-intuitive strategy for system design. It has already been recommended that the data should be captured at its source, and that only those individuals who add value to it should handle it in its forward flow. What about the paper documentation? the auditors? Conventional wisdom is to generate the document that supports the transaction at the source of the transaction. The source of the transaction is very likely to be distributed across campus or over a larger geographic area. Generating the transaction at the source requires that the forms and the required equipment (printers or other hard copy devices) also be distributed. The better approach is to capture the transaction on a terminal at the source and generate any required document or paper trail or confirmation slip at the destination and return it to the source. In this way the destination becomes the single point of document generation. This allows the destination to use high speed printing and processing techniques to generate the required document. This idea has been used very successfully by stockbrokers in transacting financial business for their customers. Electronic fund transfer systems also depend upon this method of transaction confirmation.

One important point about this approach is that signatures can no longer be the method for controlling the validity of the transaction. Since the document is generated at the destination there is no paper at the source to sign. Other methods have to be used instead of the traditional signature. Individualized passwords, magnetic credit cards, electronic signature verifications, post processing confirmations are amongst the techniques used for subsitution of signatures. More options will surely be developed as time goes on. Meanwhile, the adoption of destination point document generation can be a money saving and useful procedure.

*Online versus batch.* The economics of computing are such that online systems are more cost effective today. Terminals are getting inexpensive as are data storage devices. There is greater overall efficiency in using online procedures even though the proponents of keypunching will be quick to point out that keypunching takes less time than direct online data entry. It is, of course, true that keypunching takes less time. However, in online data entry, validity checks can be made at the time of entry insuring that data are handled once and only once. This is not true of a batch operation where error handling procedures are themselves error prone and cumbersome at best. From an overall system perspective, it is more cost effective to do as much of the process online as possible even though the data entry portion may appear to be cheaper by keypunching.

There is an ongoing belief amongst data processing professionals that some operations are inherently batch and others inherently online. This is not true. Whether a function is better performed in an online mode or batch mode depends entirely on how the operational system is conceived. Grade reporting is generally considered a batch operation, but several schemes can be devised in which this function would be better performed online. Assume for the moment that *all* students have terminals – an assumption that is not too far fetched considering that Carnegie Mellon will soon be requiring all its students to have personal computers. Given terminals, grade reporting does not need to be a batch process. Students can simply review them online without the need for the grades to be printed. As we said earlier, the rapid changes in technology compel us to examine and periodically re-examine the current processes to insure that we continue to make effective use of the tools available to us.

*Operational responsibility and control.* The computer is a tool, no more special than a telephone or a typewriter. It is different – but no more special. If it is to be viewed as a tool, it should be used as one. Administrators should integrate this tool into their operations just as they have integrated other general purpose tools into their operations. The day to day operating responsibility of the system should rest with the office charged with carrying out the function. Thus, the registrars office should run the registration system, the accounting office the accounting system and so on. It is entirely too easy in most organizational structures in existence today to blame the computing center for missing grades or late checks or other mishaps. It is no longer necessary for the institution's service departments to delegate computer related operational responsibilities to the administrative data processing group. With the increasing complexity of our information systems, knowledge of the application area has become at least as important, and perhaps more important, than knowledge of the computer system. Thus centralizing the operations function of administrative information systems is no longer a satisfactory or cost effective mode of operation.

The decentralization of computer talent into the operational areas can only be achieved if the systems are designed with this objective in mind. The systems have to be designed on the assumption that those individuals having responsibility for their operation will have no knowledge -- or at best a very minimal knowledge -- of computers and computing. This assumption necessarily results in the development of systems that have very clear man-machine interfaces. Such systems will ultimately result in the total decentralization of the administrative data processing group.

*Simplicity*. Simplicity is the key to the development of successful MIS. Keep the data structures simple, keep the displays uncluttered and simple, keep the process simple. Build complexity where needed through the use of numerous interconnected simple components and concepts. One of the major problems in this area has been the data base management system itself. The DBMS in the market today vary in degree of complexity but are all quite complex. IMS is the most popular amongst DBMS and perhaps the most complex to use. Data base systems currently in use are based on either the hierarchical or the network models. There is a simpler, easier to understand and more elegant model, due to E. F. Codd, called the relational data model.[20] The relational model consists of data represented in tables in normalized form. Originally Codd defined three normal forms of data and named them first (1NF), second (2NF) and third (3NF) normal forms. These normal forms help insure that the tables are reduced to their smallest and simplest components without any loss of data or associations amongst the data items. Subsequently, Fagin defined the fourth

---

[20] Codd, pp. 377-387.

normal form (4NF).[21] The great advantage of the relational model is that data is represented in tabular form; it is easily understood and is very easy to manipulate. All data are represented in the exact same form so that there is no increased complexity with increases in size or volume.

There are at least two commerically available DBMS packages that are based on the relational model. The first package Oracle, (Relational Software Inc., Menlo Park, California), became publicly available in July of 1980. A few months later, in December 1980, Ingres (Relational Technologies, Inc.) became available. It appears that Oracle is a spinoff of IBM's relational project called System R by industry watchers. More recently, several hardware vendors have announced their own relational data base systems. The relational data base is an emerging technology that has great potential if the associated performance problems are worked out. One major challenge facing the profession will be the migration from existing systems to a relational data base system.

At present, relational data bases offer the only hope of recovering from the losing battle of software development. Because they are easy and simple to visualize, they carry the potential of user developed software. If this does materialize, it will have a revolutionary impact on the use and users of computers.

## The Data Digraph

The data digraph is a model for the design of online data access systems.[22] The model is based on representing all the data as an interconnected graph. Graphs are diagrams consisting of nodes and arcs. If the arcs have directional orientation then the graphs are called *digraphs*. The smallest unit of data in a data digraph is a screen, up to 16 lines of data that are logically linked together and displayed as a unit. A node of the digraph consists of one

----------------

[21] R. Fagin, "Multivalued Dependencies and a New Normal Form for Relational Data Bases," IBM Research Report RJ 1812 (July 1976).

[22] This section is based on some ideas used in design of a library system described in "Design Principles for a Comprehensive Library System" by Tamer Uluakar, A. Pierce and V. Chachra, *Journal of Library Automation* 14 (2 June 1981): 78-89.

or more screens. The arcs in the digraph are the online commands that permit the user at one node to access data in another node by moving to it.

The implementation of a data digraph requires a set of online commands. All online commands are either global or local. Global commands may be entered at any time for any reason. A local command is local to the node or screen currently being displayed.

Global commands are of two types: search commands and processing commands. Processing commands initiate processes like drop courses, add courses, delete name, change security, set parameters, etc. Search commands display a screen of information that satisfies the search argument.

Local commands are of three types: traverse commands, paging commands and transactions. Assume a global search command is used to display a required screen of data. The local traverse commands then take the user to other related screens of data allowing the user to access a network of screens belonging to different nodes. Traverse commands there take you from node to node. Paging commands make up the temporary local filing system for the user. There are three commands that are recommended. These are NS (for next screen), PS (for previous screen) and show (for displaying the present screen). Each time the amount of data to be displayed exceeds the space on a screen it is broken down into pages that are accessed by using the NS command. The PS command acts as the local memory and allows the user to go back along the search path either to refresh his memory or to choose alternate search direction. The show command displays the current screen and is very useful after the "help" command. It is also useful after data entry when the present screen is no longer in view. Transactions activate routines to delete, add or modify the data of the currently accessed screen. Systems designed with this concept are as easy to use as a highway map, which they were designed to simulate.

# 5

# OPERATIONS MANAGEMENT

"Here's a good rule of thumb;
Too clever is dumb."

Ogden Nash

## Scheduling

Every multi user computer system has several levels of implicit scheduling algorithms. Peripheral devices (printers, disk, tape etc.) must be scheduled for access in multi programming environments. Virtual storage systems must schedule real storage for occupancy by applications for discrete time slices or establish some priority of interrupts. There is always an implicit scheduling of application to be processed by the Central Processing Unit.

All these implicit schedules are quite explicit to the designers of the control software. In fact, their construction is one of the primary concerns in the design of the operating system. The overall processing efficiency of the computer system is dependent on their "intelligence". Poorly conceptualized scheduling algorithms can even create lock-outs which will bring processing in the system to a standstill. In virtual storage systems, poor dispatching or swapping algorithms may lead to "thrashing" which so magnifies the system overhead that productive work on applications drops almost to zero.

The vendor does not provide custom tailored scheduling to each customer, but rather develops default scheduling in terms of how he perceives the system will be generally used, and provides "hooks" for the user to access and modify the default algorithms. Frequently, these "hooks" appear in the "job control language" where provision is made for the user to override default scheduling with explicit run time scheduling. Many of the defaults are established at the time the system is initially installed by the installation's system programming staff. Some of the scheduling algorithms can be changed only by modifying the control software released by the vendor. This last course of action should be chosen only after careful consideration of the consequences which include potential dropping of vendor software support, non trivial complications in installing new releases of the operating system and a significant on going software maintenance problem.

The most common and conceptually simplest way to schedule is by priority. Each application or task of the application is assigned a priority, that is, a numerical value. Generally, the larger the number, the higher the priority. These numerical values are ordinal and a priority value twice that of another doesn't indicate that it is twice as important, only that in an ordinal sense it is more important than any task with a lower priority.

The nature of the systems hardware is such that we generally distinguish between the job or application priority and the task priority. Any job comprises many tasks, reading tape, sending lines to the printer, acquiring real storage, accessing registers, etc. To optimize the utilization of the hardware resource is not the same as to optimize the turnaround of jobs. Consequently we have a situation of complex and interacting trade offs. Our first and probably most difficult task is the definition of priority and the goals we wish priority scheduling to accomplish. These goals will be installation dependent and should reflect the information systems philosophy and management perceptions of the total organization. That is, they probably should not be set by the Information Systems Department alone, but by the management structure of the university in concert with the Information Systems Department.

If the Information Systems Department, or at least some significant segment of it, is seen as a job shop then production scheduling will be useful. In many cases the production schedule may be most usefully analyzed by treating it as a project network and using critical path techniques. In any case, it will be necessary to identify the jobs (tasks, activities) that make up the production schedule.

Depending upon the work cycle in the Information Systems Department (and whether it operates on a seven day week, three shift day basis), the length of a project schedule will be established. The schedule is typically made up for a week although some operations prefer a monthly schedule. All of the activities to be operated upon during the schedule must be enumerated. This generally is expedited by naming the generic activities (payroll, accounts receivable, purchase orders, etc.) and sub-numbering the activities in the generic chain. The totality of activities required to process the payroll might be PAY101 through PAY123 — 23 discrete activities involved in processing the payroll.

We need next to establish the precedence requirements for all activities, i.e., which activities must precede this activity, which may not start until it is completed, and which may be executed independently of the execution of this activity. In many cases this analysis will expose all manner of flaws in the design of the system, or at least in the modifications that have been made since it was initally developed.

At the same time we need to identify those resources making up our job shop. Such resources might include data entry (both input and verification), run control, CPU processing, printing, breakdown, distribution, etc. Each activity in our schedule must be tagged as to the resources it requires. Additionally, we must estimate the quantity of each resource available during the time period of our production schedule. Realistic estimates of resource availability will be sensitive to equipment down time, operator effectiveness (nobody actually works eight hours in a 9 to 5 day), etc.

With both the activity precedences and the resource quantities determined, we next proceed to estimate the resources consumed by each activity as it is processed. If this information is not currently available it will need to be developed by tracking the activities through several cycles of the production schedule. We need to be careful not to be overly optimistic as much as to avoid always planning for the worst. Keep in mind the point that the schedule is a plan, not an observed fact. The schedule is not immune to slippage, but should help pinpoint critical activities in the processing chain and more clearly show the future impact of current slippage.

## Critical Path

Frequently, the results of the initial production schedule are quite perverse. We discover that in order to make our deadlines we need to use more of a particular resource on a given day than is available. Often we discover that resource utilization is quite variable through the period of the production schedule. This leads to peak load staffing and significant underutilization of resources in order to permit meeting deadlines.

What we need is a strategy for load balancing which will have no (or minimal) impact on deadlines. If we wish to create as uniform resource loading as possible we must be prepared to adjust the time when activities are processed, subject to not slipping the deadlines that establish the schedule. This is possible only if there exists slack time in one or more of the activities making up the schedule. An entropic view of life will convince us that there is no such thing as a schedule with no slack and the corollary is that there is no schedule that uses 100% of the resources.

The problem can be attacked by constructing a daily resource utilization plot or table for the period of the schedule for each resource. This plot will typically show peaks and valleys — no peak can be larger than 100% or the schedule is not feasible. A relative measure of the resource utilization balance is the sum of the squares of the daily resource utilization for the period of the schedule. For instance, suppose we have a three-day schedule with resource utilizations of 5, 11 and 8 on the three days. Then (5x5 + 11x11 + 8x8 = 210) is a measure of the resource utilization balance. Any schedule with a lower sum of squares is a superior schedule in terms of balancing the resource loading. For instance, a schedule of 6, 10 and 8 (6x6 + 10x10 + 8x8 = 200) is better and a schedule of 8, 8 and 8 (192) is best of all. With the criteria of minimizing the sum of squares, we could begin shifting activities around so as to achieve better load balancing. However, we run the risk of balancing resource X at the expense of greatly imbalancing resources Y and Z. As the interactions become more complex, it probably behooves us to find a more formal technique which can be written as a computer algorithm.[23]

---

[23] Joseph Moder and Cecil Phillips, *Project Management with CPM and PERT* (New York, N.Y.: Reinhold Publishing Corporation, 1965).

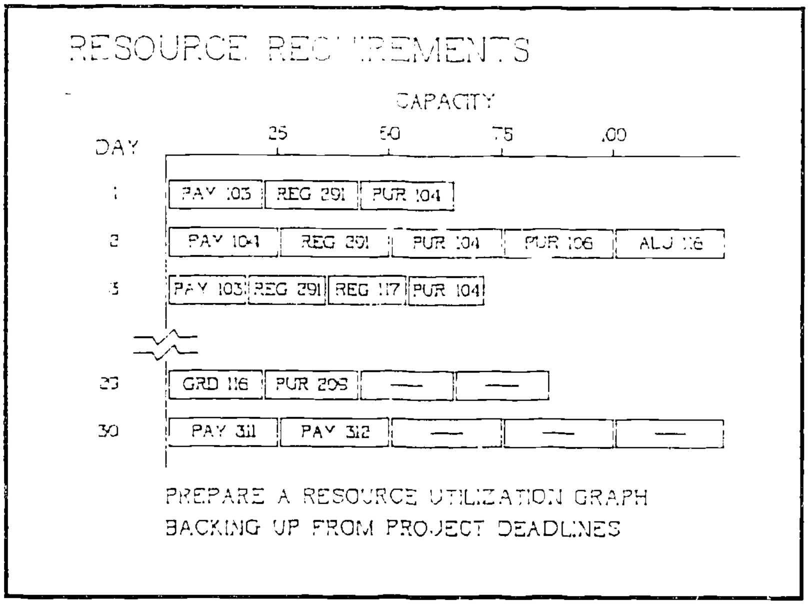



Figure 34

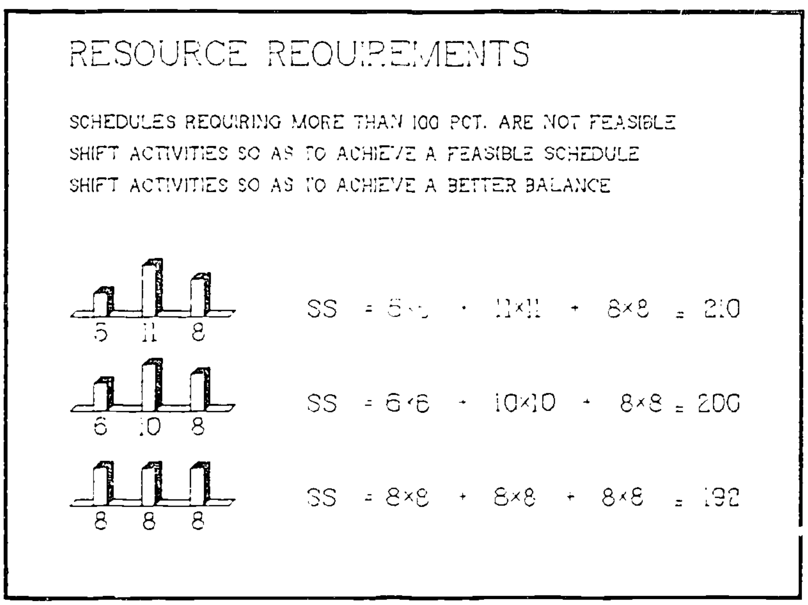



Figure 35

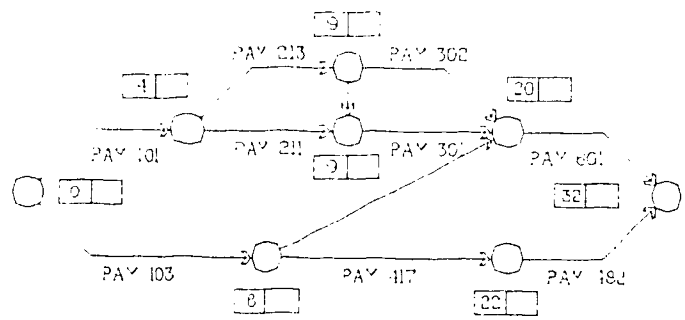
Many scheduling and resource allocation problems are usefully represented as networks. Network analysis is part of the formal structure of mathematics, subsumed in the area known as graph theory. We will content ourselves with a more specific look at networks as they relate to project management. Any project may be considered to be made up of uniquely identifiable activities, the sum of which constitute the project. An entire business might be considered a project, or the business might be thought of as the sum of a number of projects. The division is arbitrary and selected for the convenience of the analyst. Similarly, the activities could be composed from sub-activities and so on. It will suffice for our initial purposes to look at only two levels, a project and its constituent activities.

| Activity | Description |
|---|---|
| A | Keystroke time sheet data |
| B | Keystroke master account file updates |
| C | Run account breakouts |
| D | Run time sheet exception reports |
| E | Print weekly account summaries |
| F | Run overtime compensation report |
| G | Run master file update |
| H | Run exception report on master file |
| I | Prepare new master file for A/R system |
| J | Run payroll checks |

The activities making up the project may be thought of as a graph with nodes, representing the end points (milestones) of activities and directed, connecting lines representing the activities themselves. The resulting graph then shows the precedence requirements of the network. Consider the following description of a network for the activities shown above.

| Activity | Nodes | Schedule Requirements |
|---|---|---|
| A | 1-2 | May begin with the project |
| B | 1-3 | May begin with the project |
| C | 2-5 | May begin after A completes |
| D | 2-4 | May begin after A completes |
| E | 5-6 | May begin after C & D complete |
| F | 4-6 | May begin after D completes |
| G | 3-6 | May begin after B completes |
| H | 3-7 | May begin after B completes |
| I | 7-8 | May begin after H completes |
| J | 6-8 | May begin after G, F & E complete |

# PROJECT NETWORK
# CRITICAL PATH METHOD




| ACTIVITIES | NAME | TIME | SCHEDULE |
|---|---|---|---|
| KEYSTROKE TIME SHEETS | PAY101 | 4 | ---- |
| KEYSTROKE MASTER UPDATES | PAY103 | 6 | ---- |
| RUN ACCOUNTS BREAKOUTS | PAY211 | 2 | AFTER PAY101 |
| RUN TIME SHEET EXCEPTIONS | PAY213 | 5 | AFTER PAY101 |
| PRINT WEEKLY SUMMARIES | PAY301 | 10 | AFTER PAY211 AND PAY213 |
| RUN OVERTIME COMPENSATION | PAY302 | 11 | AFTER PAY213 |
| RUN MASTER UPDATES | PAY416 | 8 | AFTER PAY103 |
| RUN MASTER EXCEPTIONS | PAY417 | 16 | AFTER PAY103 |
| PREPARE NEW MASTER | PAY482 | 6 | AFTER PAY417 |
| PRINT PAYROLL CHECKS | PAY601 | 12 | AFTER PAY301, PAY302 AND PAY416 |

Figure 36

The project is completed whenever the last activity [illegible]. In this case the project is processing the payroll and the constituent activities are ten in number. The start and end nodes of each activity have been numbered so that we could refer to activity A as 1-2, activity G as 3-6, etc. An activity, [illegible] must be drawn in order to meet the precedence requirement for activity [illegible]. This will be called a dummy activity which, although not a stated activity in the project description, is nonetheless necessary to graphically show precedence. This description of the network we will refer to as its topology.

In addition to the topology of the project, each activity will have one or more characteristics which we will call its geometry. For our example network, we may know the time required to perform each activity. This characteristic could be [illegible] to our network description matrix.

Typically, we will want to ascertain the time required [illegible] the milestones in our network. The geometry of the network [illegible] include other activity characteristics such as cost, probabilistic descriptions of time and/or cost, other resources used by the activities, etc. Each of these geometries may also be studied in the network analysis.

If we distinguish between deterministic and probabilistic geometries, there are two well known techniques of analysis. If the geometry is deterministic the technique generally used is called the Critical Path Method (CPM). If the geometry is probabilistic we use the Program Evaluation and Review Technique (PERT). Each analytic procedure presumes some objective function, such as minimum time to complete the network.

The simplest case is CPM where both the geometry and topology are deterministic. Picking some time (say T=0) as the initial time, we wish to determine the earliest times for all milestones to be reached; the latest we will call the early completion time. Our precedence graph requires that no activity begin until all its predecessors have been completed, the longest path through the network being called the critical path. The early start of the nodes is easily calculated.

Given some target completion time, we could determine the late finish times for each activity so as to meet the target time. This could be accomplished by creating a "dual" of our early start function to process backwards through the network.

It is frequently useful to talk about the slack or float time of activities. The total float is defined as the difference between the late and early starts (or the late and early finishes) of the activities. It is observed that the total float of a number of activities is zero indicating that no slack is available; that is, they are on the critical path. We could also look at float in terms of either the early start or late start networks.

An objection raised to CPM has to do with the activity times.[24] To presume they are deterministic is to assume that *a priori* knowledge (usually in the form of experience) is available. In computer operations, for instance, we might be willing to agree that experience in running the payroll allows us to state with great confidence the CPU time required to run the next payroll. On the other hand, the software designer might argue that each of his projects (and in most senses, its activities) represent new, unique undertakings and a probabilistic estimate is more appropriate. Any appropriate statistical distribution could be assumed and its expected value used as the time characteristic in a CPM analysis. Because most estimates are made by people not particularly familiar with statistics, a simplified process based upon the character of the Beta distribution has been developed. The estimator is asked to suggest values for an optimistic, a pessimistic and a most likely time. The expected time and its standard deviation are then estimated as the sum of the optimistic, pessimistic and four times the most likely, all divided by six, and the pessimistic minus the optimistic divided by six.

A time analysis of the network is then performed with CPM using the expected times computed as above. If we assume that any path is a chain of independent activities, we may take the variance of the path to be the sum of the variances of individual activities making up the path, the mean time for the path to be the sum of the activity expected times, and the distribution of the path time to be Gaussian.

We may now deal with questions such as "What is the probability the project will take more than X time units?". One must remember that we have considered only the critical path. We could also investigate alternate paths through the network.

---

[24] Ibid, p. 135.

Probably the most obvious resource constraining our production schedule is time. The time constraint, expressed as deadlines, was used implicitly (by working backward) to establish the schedule originally. In CPM terms, the only way to improve the schedule is to shorten activities on the critical path — all non-critical activities have some float. We can generally think of some way to expedite any activity or set of activities. In fact, this generally is the underlying reason why we acquire a larger CPU, add data entry personnel, etc. The process of shortening the critical path is called crashing.

Crashing infers the need for some kind of cost trade-offs. We usually pick the method of processing an activity which is cheapest for that activity. This is a form of sub-optimization as we never asked how this particular activity impacts the cost of the whole project. An alternative method of processing an activity may allow it to be completed faster, but at greater cost. We could use this expedited processing, complete the schedule faster, at some known incremental cost. This cost could then be compared to other alternatives for expediting the schedule (and their costs) to determine the least expensive way to improve the schedule.

The more difficult type of resource constraint occurs when the resource is fixed in size. Multiple activities contending for the resource may require that some go unprocessed, or at least be deferred for processing, which drags out the schedule. There exist mathematical algorithms which will assist in developing the least dragged out schedule. They are somewhat beyond our general level of discussion but are imbedded in most CPM algorithms implemented on computers.

## Service Charging

Scale considerations and organizational dynamics have led many universities to conclude that there are significant economies of scale in computing installations as well as desirable organizational efficiencies experienced by centralization of responsibility for the total computationalfunction.

For reasons previously discussed, this centralizing move hardly ever results in a single large scale computer for the satisfaction of the total university information systems needs. Far more likely is the development of a central authority controlling a local network featuring multiple CPUs, multiple I/O devices and generally an array of local teleprocessing terminal devices.[25] For many administratively or geographically decentralized universities (or for economy minded non competitors) some form of remote teleprocessing network usually augments or adds to the local network.

Such organizations are faced with the same plethora of management problems as are many of their customers. Not unique among them are problems of physical and personnel inventories, scheduling algorithms and the necessity of accountability if only for next year's budget justification. What is needed by such computer authorities is a management information system.

Since the data elements inherent to the decision processes are by and large the service items being purveyed, and since the device purveying the bulk of the service items has a memory and is constantly monitoring itself, an obvious conclusion suggests itself. Why not cost the items of service and operate as a "cost" or "profit" center, thereby generating inventories and audit trails of services provided, building in appropriate scheduling techniques and directly addressing the accountability information problem?

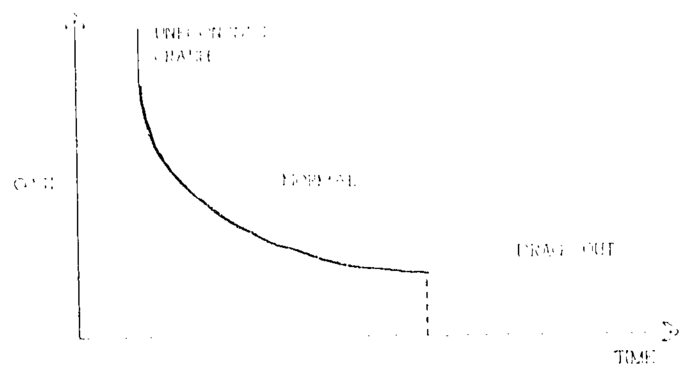
Many would argue that the overhead and complexity of operation as a cost center is too high for smaller centers and consequently an unwise and uneconomic decision. Others would rebut that a system so configured must then be inefficient (and uneconomic) from the standpoint of economies of scale. For our purposes here, suffice it to say that computer services charging is generally efficient and desirable for medium to large installations and smaller installations could be so accounted if they subscribed to a network containing at least one larger installation.

For any size installation, if service (in the computer utility sense) is not the primary motivation for the computer operation, computer services charging will have less desirability. The typical comparisons drawn between computing centers and production lines, libraries, or power utilities might make a strong case for stipulating that all central computer operations should be in the utility or job shop mode and single user minis in the library mode.

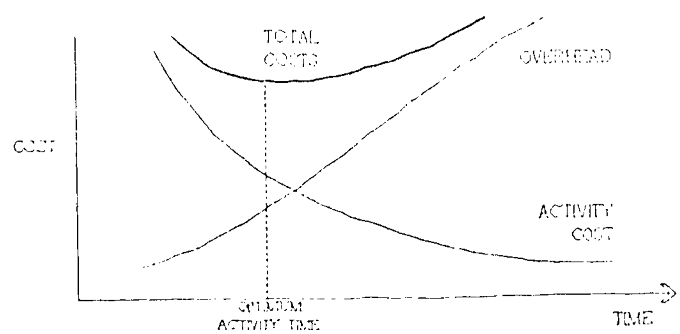
---

[25] This section is adopted from R. C. Heterick, "Systems Management and Allocation of Resources Technique," *Economic Considerations in Managing the Computer Installation*, Association for Computing Machinery Proceedings, 1971.

## RESOURCE CONSTRAINED NETWORKS




FOR ANY NETWORK ACTIVITY WE CAN IDENTIFY ALTERNATE SCHEMES FOR ACCOMPLISHING THE ACTIVITY. THERE IS A COST PENALTY ASSOCIATED WITH EXPEDITING THE ACTIVITY.




IF THE EXPEDITING COST IS OFFSET BY REDUCED OVERHEAD COSTS, THEN IT IS COST EFFECTIVE TO CRASH THE ACTIVITY.

Figure 37

Given a computing system configured somewhat as described, what must computer services charging accomplish? This question is probably best viewed in terms of the three most involved constituencies: the university administrative hierarchy, the users of the computing center and the management of the center itself.

A computer services charging scheme implies a money management and distribution system — an economy. Other forms of "computer credit" systems have been devised, but the use of a dollar system remains prevalent probably because it is more comprehensible to administrators who may or may not understand computer systems jargon. No one in the computing user communities will be without an understanding of a dollar charge, as this form of money management is exactly what they are accustomed to in their operating, capital outlay, and other budgets. In fact, for some departments in the organization, the computer budget may exceed the total of all other budgets. Consistent with university policy in managing other budgetary items, computing dollars should be seen as liquid assets, exchangeable for services other than computing on a dollar for dollar basis.

This is, of course, one of the major rationales for management to consider computer services charging. To satisfy fully money managers in the administrative hierarchy, the distribution and control of computer services funds should conform as nearly as possible to the administrative hierarchy itself. The implication is that he who gives can also take away.

In such a budgetary system, all accounts point to a funding account with no particular hierarchy assumed. Consequently, the funding and accounting structure embodied in the job accounting system parallels exactly the organizational management structure. Embedded within this administrative accounting structure should be some provision for extracting program related accounting data. As higher levels of management aggregation are reached, program rather than administrative budgeting may become more attractive or rational. Only end users may actually purchase computer services with their funds. All other accounts exist solely for the purpose of money management and accounting. Competition for credit allotments to purchase the limited computer resource is monitored by the various levels of administrators within the university. Administrators are able to control the allocation of funds in all accounts directly funded by them (shifting, adding to, or reducing funds) and indirectly influence all accounts in their chain of authority.

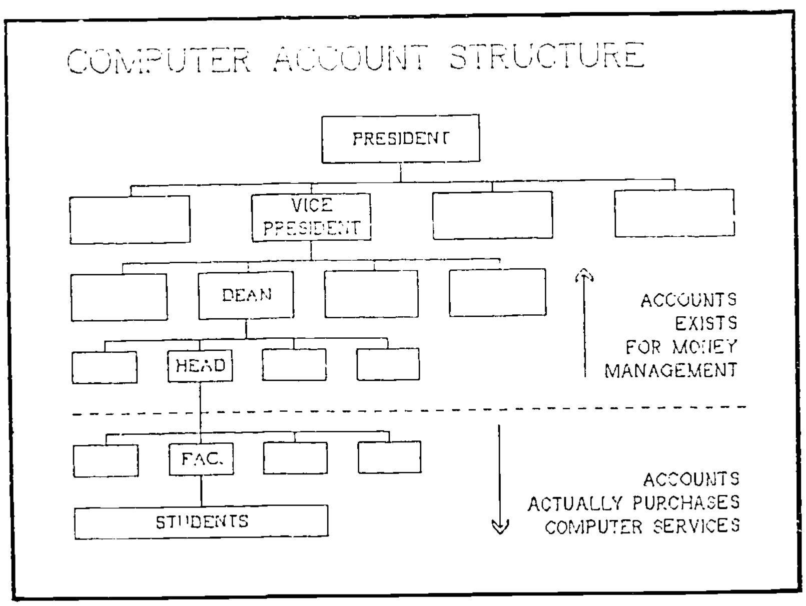

COMPUTER ACCOUNT STRUCTURE
PRESIDENT
VICE PRESIDENT
DEAN
HEAD
ACCOUNTS EXISTS FOR MONEY MANAGEMENT
FAC.
STUDENTS
ACCOUNTS ACTUALLY PURCHASES COMPUTER SERVICES

Figure 38

Computer services charging to be successful must be realistic, equitable, and imposed to recover costs (and profit) rather than arbitrarily discriminate service classes. Realistic implies pricing to recover both direct and indirect operating costs. Equitable involves appropriately defining units of service in terms of differential system resources. The user is much concerned with equity but only slightly with realism. The center management, on the other hand, is greatly concerned with recovering total operating costs. Charging to recover costs satisfies auditors and, generally, internal operating divisions within the university structure. Arbitrarily discriminating services classes will likely create problems with government auditors and possibly with internal auditing staffs.

We are often bemused by the information systems manager who points to his low rates as justification for service levels achieved. Frequently, indirect costs such as floor space, custodial services, air conditioning and electric service, etc. are not figured into the rate structure. In fact, if most universities appropriately evaluated the cost of floor space (particularly on urban campuses) we would expect to see the computing center somewhere in the far environs of the campus rather than in the main administration building. It is not uncommon to see equipment costs amortized over a fifteen-year period when the economic life of the CPU is hardly ever more than three years and most peripherals seldom more than six.

Given typical budgeting procedures followed in computational intensive organizations (universities, research and development firms, financial enterprises etc.), it is likely that the data processing budget is little related to the potential demand for computing. Where there exists some external demand for services, an arbitrary multiplier (greater than 1.0) of the internal rate might be applied.

The user wishes to have a technique whereby he can influence the allocation of computer resources to his jobs. Computer services charging admits of this influence within the constraints of total funds available to the user. Charging removes the arbitrary constraints of job classes, time limits, storage constraints, and the like. It imposes the alternative burden of money management on the user. Experience indicates the user prefers the money management problem as it focuses his selling job on his administrative superior rather than on those "unknowledgeables" in the computing center. This shifting of managerial responsibility back to the administrative structure is regarded somewhat ambivalently by administrators. Strong administrators welcome it; many regard it as a mixed blessing. The ability to influence turnaround time is a necessary prerequisite for user acceptance. It additionally contributes to improved load balancing.

From the standpoint of the computing center itself, computer services charging presents a confusing dichotomy. While gratified to be relieved of scheduling and resource allocation problems (which they are probably not well qualified to make), efficiency of utilization of the computer resource is also removed from their direct control. A high level of understanding and communication between center management and top management of the university is essential. Top management must be convinced that something approaching 100% efficiency of computer resource usage does not imply maximum resource allocation to optimizing the university objective functions (whatever they are). Parallels between the use of the motor pool, office space, etc. must be drawn and understood. The mystique of 100% utilization must be dispelled, or center management is confronted by unresolvable conflicts that will obviate the advantages of cost center operation.

## Job Accounting

The cost center concept is most easily implemented by a service utility oriented computing organization. This service orientation implies a management that seems to be increasingly typical of installation managers in the large corporate and institutional computing complexes. An installation management that views itself as providing information systems services and considers the internal design of software and operation of hardware as secondary readily accepts the cost center concept. Centralized "make or buy" decisions immediately raise the questions of in-house vs. external costs in the satisfaction of organizational computing needs.

These cost trade-offs are meaningful only if the center is aware of its "real" costs of operation. For those centers not operated as a separate corporation, the overhead (electricity, physical plant, external administration, etc.) attributable to the center must be distributed to it and made an integral part of its pricing and "make or buy" decision process. Those organizations that encourage the dispersal of data preparation, program submittal, and output checking back to the operating departments should be capable of developing more realistic and equitable charge rates. For highly decentralized organizations such as universities, this overhead distribution may be 20% of the actual hardware budget. For more centralized organizations, these overhead costs may be 100% or more of the hardware budget. In fact, for highly centralized computing operations, the ascertainment of "real" operating costs may lead to strong economic arguments in favor of a more decentralized type of operation.

The funding of a cost center operation can present difficult problems to computer center management. It is doubtful whether a cost center approach will work if "make or buy" decisions are not vested solely in the installation manager or information systems coordinator. The additional elasticity of demand (and its ancillary cash flow problems) generated by permitting non-information systems administrators carte blanche to decide whether to buy inside or out, is a formidable problem accompanied by an absence of publicized successes. Since the computing capability is constant (at least in the 90 to 180 day short run) in the face of a variable demand, probably the best funding procedure is an "off the top" minimum pledge by all operating divisions of funds which will be transferred to the center, irrespective of lesser demand. The elasticity of sales and hardware lead time problems are thereby alleviated. In their place is substituted the necessity for the center to predict accurately the number of units of chargeable items to be sold to insure reasonable prices and adequate cost recoveries. Under or over recoveries of 10% should present minimal problems. Deviations of more than 10% can create serious dislocations. In order to avoid the problem of under or over recovery, suggestions have been made to engage in retrospective charging based on service units actually supplied during a week or month. This would of course be totally impractical for commercial time sharing services and seems an unfair abdication of center management responsibility in noncompetitive situations. The user should not be faced with variable costs, totally beyond his control, simply to relieve management of its planning responsibility. A technique of anticipatory rather than retrospective rate setting seems more reasonable.

In a cost recovery, service charging environment, user liability is or can be a serious concern. The most effective way to limit user liability for unauthorized use of his account is multiple levels of account protection. These levels of protection can range from routing output through locked boxes to typical account password protection.

Additionally, the account structure may be established on a credit balance principle. The user is liable only for the amount of money in his account and any attempts to process a job generating charges in excess of his account balance will be automatically flushed from the system.

It is worth mentioning that charges must be reproducible to provide suitable user equity. As in the case of retrospective charging, the user should not be subjected to variable processing costs beyond his control. Sufficient differentiation of service charges should negate most of the arguments advanced for defraying variable system overhead to user jobs.

The cost center approach implies a formalism for handling programming and systems analysis that many universities do not currently follow. As charges are made for these services, allocation of the programming resource becomes highly affected by user demand. The technique that seems to have been universally accepted to minimize this impact is the RFP (Request for Proposal) procedure. All non trivial requests for program changes or systems development generate a proposal from the center to the requestor with time, personnel, and cost constraints indicated. This proposal is negotiated until such time as there is mutual agreement upon a statement of effort.

Many of the currently extant accounting packages do project rather than job accounting. This is particularly true of the batch packages. Project accounting represents an aggregation or accumulation of job accounting information but in practice cannot substitute for job accounting. If the user is to assume the burden of money (machine resource) management, he needs a "real time" bill for services associated with each job processed. This is particularly the case in highly decentralized programming environments.

Once an installation has embarked upon a course of computer services charging to facilitate better resource allocation, it is confronted with the question of how much and for what to charge. Given the architecture of most current hardware and typical system usage, the following categories of charge are generally settled upon:

Central Processor
Unit Record Devices
Setup Devices
Storage Media
Priority
Personnel Time
Communication Support

Software
Rental Equipment
Maintenance

In general, operating overhead is distributed to these categories in some reasonably equitable fashion. Operating overhead would include center administrative salaries, distributed organizational overhead, and all center services which are provided at no direct charge to the user. Such center services frequently include in-house consulting, general user publications, and contributed staff time to extra-center organizational committees and tasks. For the typical multi programmed system, central processor charges are generally made in three categories (i.e. execution time, I O events, and core time). Unit record charges are generally made for cards read, cards punched, lines printed, and plot lines or time. Setup device charges are usually imposed for use of demountable tape and disk units, special forms and print trains, special plotter setups, etc. Storage charges are imposed on tape and disk files (generally to include recovery of library maintenance costs).

Some form of a priority of service charge needs to be implemented to develop user acceptance of the cost center technique. For most users, turnaround time will be the most important resource they can purchase. Several alternatives are available. Cost differentials can be associated with prime or non-prime shifts, or a hierarchy of first in-first out (FIFO) job queues may be supported. The FIFO job queues is by far the simplest method to implement and easiest to support. Turnaround time itself is not sold, but rather relative position in the job queue is purchased. Personnel charges for programming and systems analysis are generally straightforward and easily arrived at. It seems generally preferable to build operations personnel costs into appropriate hardware service categories. It is generally necessary to recover communications control device costs through a line connect time charge. Remote batch computing charges can be computed in the same manner as local batch. Timesharing users should probably be charged for the same categories as batch users, but at a relatively higher (or the highest) priority rate. Software charging presents a new dimension to computer services charging. Many installations recognize proprietary software costs of from 10% to 25% of hardware costs. The simplest approach to software charging is on a per access basis. For many installations growing slowly into online management information systems. such a charging scheme may need to be augmented by some form of subsidy until a sufficient number of users are online, to create more reasonable charge rates.

## Auditing

The actual determination of unit prices can be quite complex in practice. A general approach is to determine a unit price based upon annual service charge (ASC) divided by expected annual demand (EAD). The annual service charge is computed as the sum of direct hardware (lease or amortized value) plus indirect hardware software plus direct personnel plus percentage of total overhead.

On the basis of past data and some extrapolation of changing demand, the EAD is determined. The unit charge is then determined and rounded if desired. Rates are generally not adjusted more frequently than every twelve months to provide some stability for users in their money management.

While the operating system is reading a job submitted to the system, certain job accounting routines verify that the job's account number is eligible to use the system and that this account has funds to pay for the requested processing service. This information is available from a job accounting disk file containing records for each account holder. If funds are not available the job will be flushed and not processed. If the job passes all initial tests, it will enter the appropriate job queue or go immediately to processing if it is a timesharing job. After the job has been processed, other job accounting routines collect all billing information from the operating system, calculate the associated charges, and update the proper account record. The final page of printed output can be a detail bill showing the current balance before processing, all costs for computer resources, including a surcharge or rebate for the selected priority, and the current balance after processing. A bill for each job run is a necessity if the user is to manage wisely those computer funds allocated to him.

Each account record might have several hundred fields of information. The job accounting routines update the data in all appropriate fields. All of these fields of information can be available for immediate access by a terminal (a typewriter or CRT terminal in the center administrative offices) and for any report generating programs in batch mode. The terminal, with its own command language, can be used to open, close, refund, adjust, and query accounts as well as handle fund transfers from funding accounts to user accounts and vice-versa. The report generating programs are principally used to create end-of-month reports for the university management structure, computing center administration and individual account holders.

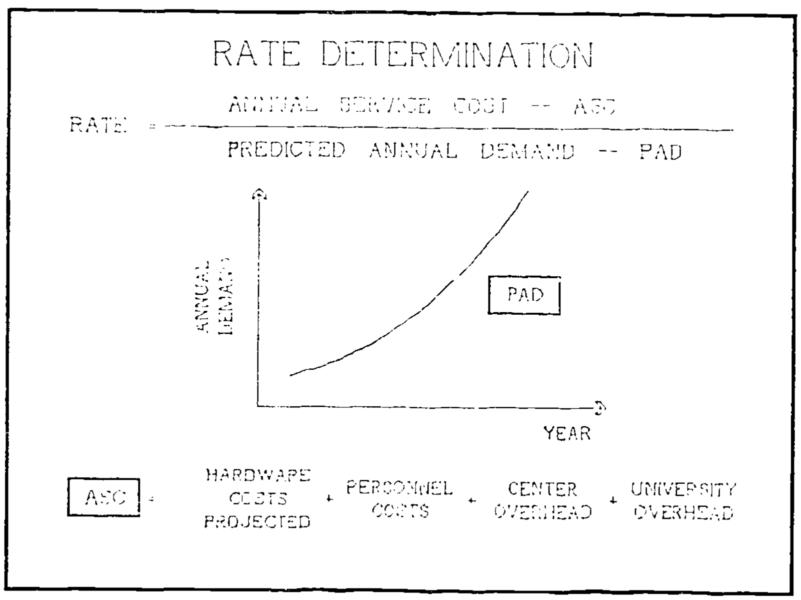

RATE DETERMINATION
RATE =
ANNUAL SERVICE COST -- ASC
PREDICTED ANNUAL DEMAND -- PAD
ANNUAL DEMAND
PAD
YEAR
ASC =
HARDWARE COSTS PROJECTED
+
PERSONNEL COSTS
+
CENTER OVERHEAD
+
UNIVERSITY OVERHEAD

Figure 39

Computer services charging places several additional responsibilities upon the computing center. Specifically, the center must provide daily audit trails and reasonably extensive security provisions. Under such a system, when a user wishes to open an account with the center, he submits a form indicating the funds be allocated to him signed by a valid funding account holder. By stating the amount of funds allocated, the user establishes an upper limit of liability for which he may be billed. Upon opening of the account, the user is assigned an account number and a post office box number. The box can be a typical post office box with a key lock. Post office boxes can be located at the center and in remote stations scattered throughout the campus. The user submits his job with a job name that designates the station and box number that his output should be routed to. This eliminates the need for operators to collate input with output prior to returning it to the user. The user may protect his account by referencing his account number to his station-box number. While this does not preclude others from using his account number, it does prohibit them from receiving the job output, thereby eliminating any advantage of using another account number. Additionally, the user may create and change at his discretion passwords for his account number. Failure to use the correct account number/station-box number/password combination will flush the job from the system. Numerous other software/hardware constraints may be imposed by the user on his account. Jobs can be time, storage, total cost, or hardware resource limited at the user's discretion.

Extracted summary data from the job accounting file may be kept in an online installation management system available for query from a terminal. Typically, summaries of accounting information by remote site, program budgeted categories, total cost recoveries, etc., may be maintained. The job accounting file may be used to monitor continuously recovery rates and to review macroscopically the adequacy of the rate structure.

The primary variable being optimized by the priority feature of a job accounting system is turnaround time. By selecting his own priority, the user can control turnaround time to the extent he feels he can afford to pay. However, designing strictly for turnaround time can greatly affect throughput and system utilization. Some techniques can be used to ameliorate this conflict. In order to accomplish a better balance of user turnaround time and system utilization, the priority scheduler could perform two auxiliary functions called job aging and pass-over. Jobs in lower queues can be aged, or bumped up, to a higher job queue after residing in the original queue for a fixed amount of time and not yet selected for service. A job selection algorithm first attempts to select jobs for processing from the highest priority queue containing jobs. Selection is by first in, first out within priority with the following exception. If the job next to be selected requires a system resource (such as storage) which is not currently available, the algorithm will pass over that job and search the queue (highest to lowest) for a job fitting into the resources available and having a job time of less than the soonest to be completed job currently active on the system. This technique allows quick running (and generally small storage) jobs to use these otherwise unused resources without noticeably affecting the turnaround of higher priority jobs being passed over. In order to provide an audit of job accounting transactions, particularly for reconciling user problems, a daily report is produced to list all detail transactions handled by the system. These transactions include not only the online transactions, but also those handled by the administrative terminal or by batch updating of miscellaneous personnel and data preparation charges.

## Queueing and Simulation

Perhaps the most direct way to attack the scheduling problem is through theory. At every point in the system at which a choice must be made as to which task occurs next, a queue develops. One could then view the scheduling problem as finding the optimal strategy for servicing queues. In order to subject a waiting line or queueing problem to formal analysis we will need to know, find out, or make realistic assumptions about the parameters of the queue.

We must identify the population of potential customers for service. Correspondingly, we must identify the one or more servers who will provide the service. If all the servers are busy then queues will begin to develop. We must know the arrival pattern of customers desiring service in the system and the distribution of times required to service customers. We expect that from this information and the mathematics of queueing theory we will be able to estimate mean waiting time in the queue or system, mean number of customers waiting for any particular service and the mean number of customers in the system.

This is hardly the place for an exposition of queueing theory, but we could investigate some simplistic applications to provide a flavor for how queueing models might help us discover good scheduling strategies. If the system has an average of two jobs per minute entering, and if the inter-arrival times are exponentially distributed (a common and usually valid assumption) there is a 40% chance that more than two jobs will enter the system in any minute. If the system is heavily utilized, say greater than 90%, we can discern that we will begin to develop very long queues and increase substantially the mean turnaround time for all jobs.

Queueing theory will also tell us interesting facts about the effects of scale. If we double the mean arrival rate and the mean service time we halve the mean waiting time but leave the mean queue size unchanged. Applying this line of reasoning to shifting our workload to two computers (each half the speed of the larger system) the mean number waiting will remain the same, but the mean waiting time will double one more weapon in the arsenal of those who favor larger machines.

The results of queueing theory problems, and some that are not capable of queueing analysis, may also be obtained through simulation. Simulation carries with it the penalty of increased problem setup time and frequently extensive utilization of a computational resource to effect the solution. There are several excellent simulation languages, GPSS (IBM's General Purpose System Simulator) being one of the easiest to learn and use.

In the terminology of GPSS a transaction (in our example, a job) must be GENERATEd, pass through the system QUEUEing for facilities (in our example, the server), SEIZEing and ADVANCEing through the facilities, RELEASEing the facility, and finally, TERMINATEing or leaving the system.

The system is structured around a clock. A time for the next transaction to enter the system is set by the GENERATE command. The current transaction is moved forward until it is required to wait, either for a busy facility or the necessity to be serviced. The status of each transaction is kept in a table showing its next time to move, where to move to, and its current location. Upon encountering the TERMINATE command the transaction is removed from the table. Each transaction may be assigned a priority and queueing disciplines are effected by the priority. We could similarly create FIFO (first in-first out), LIFO (last in-first out) or any other discipline we wish.

Perhaps the best way to gain insight in predicting system performance and attempting *a priori* analysis of the impact of system changes is through simulation. There are a number of proprietary software products on the market which will even simulate the behavior of the basic hardware components. At this level it is frequently useful to have developed a synthetic job stream of kernel routines. These kernel routines are designed so as to exercise the computer system similarly to application software for

Compute bound routines
I/O bound routines
Data movement
File updating
Compilation and execution errors
Library manipulation

and also kernels for

Data management
Telecommunications
Storage allocation
Spooling
Restart and recovery
Hardware and software protection.

Once a regimen of statistics gathering has been established one can begin to seriously consider simulation of the system. The input to the simulation model can be drawn from the data gathered for purposes of performance evaluation. In many instances, the data needed for model verification will also be found amongst the statistics developed for performance evaluation.

## Security and Privacy

One of the best ways to improve communication with both users and managers is to provide them meaningful information on the workload and the response of the computer system. Such information is of little value if not presented in terms its intended audience can understand and deal with. Like most managers, the information systems manager is somewhat dubious of an open policy regarding performance information. This potentially damaging information may, in fact, not even be available to the information systems manager.

As to the latter, it is difficult to sympathize with the manager who attempts to control and direct his operations (often with annual budget in the hundreds of thousands, or millions) by intuition and visceral reaction. Charging, scheduling and tracking all presume and require the existence of quantitative measurements of performance.

Top management is most frequently interested in who is using the computer resource, and for what. Good organization of management summaries can help make the point that there are really a number of resources involved. In this regard, it may be a good idea to keep program budgeted accounting records as well as departmental records. Nothing is more depressing to a busy manager than a report in which he must wade through twelve pages in which he gleans two items of useful information. Equally as distasteful is the crammed output form which requires "nine men and a Philadelphia lawyer" to interpret. Far too little use is made of graphs and plots as summary reports for management. Exception reporting is even preferable if one could ever get top management to stop and define the norms so that the exceptions could be identified. Management summaries are nearly always better expressed in percentage rather than absolute terms. In this respect, some modification of the Kiviat graph might be an ideal way to express a good deal of highly summarized information.

Last, but certainly not least, an effective tracking system should address security issues. We might categorize security in two ways; physical and software. Of the two, the latter is directly the province of the Information Systems Department, the former requiring a good deal of cooperation throughout the whole organization.

Physical security is somewhat easier to implement once top management gets over the feeling that it must display, usually behind glass walls, its computer operation. That may not seem like much of a concern for those of us who have never gotten out of the basement, but the attitude is more harmful than the physical exposure. Access to the machine room should be limited to operations staff and those members of the systems programming staff who are specifically needed to correct problems which have been identified by the operations staff. Tours, if one must have them, should be scheduled well in advance and carefully planned for. Even better, tours should be conducted via video tape. Entrance to the machine room can be controlled by inexpensive badge reading and combination lock devices.

Increased usage of remote batch and teleprocessing should help to strengthen the real security soft spots of input and output. In a teleprocessing environment, sensitive input forms and output reports never exist as physical items in the information systems department. Once users familiarize themselves with the ubiquity of online update and inquiry, the need for so many input and output forms is greatly reduced. For the user office, much of this data retains its former (B.C. before computers) sensitivity and is treated much more carefully in their own office.

Software security is complex and expensive, but under the total control of the Information Systems Department. Increased usage of teleprocessing substantially raises the need for software security. As noted by the vendor scramble to st[illegible] the issue, this is one of the weakest design points in most operating systems. Password control, one of the easiest forms of security to implement, is one of the least effective security measures unless treated quite seriously by all users Sensitive data in large files should be protected by the data base management system as well as the operating system. In many cases it may be worthwhile to enforce security not only by password, but by physical device address. The physical devices can be further controlled by requiring a key or special magnetic card to be used in turning on the device. One can think of other security features, such as data encryptation, but in the absence of real security concerns on the part of all users they will always be easily circumvented. A good security program must start with education and follow through with monitoring.

# 6

# INFORMATION MANAGEMENT ROLES

"If a man is willing to go as far as he can see, he will be able to see farther when he gets there."

## The User

In the section dealing with Software Development a "user" was defined as the individual who paid the bill for the systems development process. That definition is correct but inadequate. Developing a system like the registration system affects not only the registrar's office but the entire community students, faculty, staff - at the institution. We, therefore, have to broaden our definition of the user. A user must not only pay the bill but also must be an individual with sufficient management responsibility and management perspective to insure that new systems do not change institutional *policy* without proper review and approval.That is not to say that on occasion institutional policy does not merit change - it does, but such changes in policy should be properly considered and approved and not be implemented by system default. Notice that so far we have said nothing about procedures. Innovative systems design will indeed require that, from time to time, operating procedures be revised or changed altogether. The user must have sufficient administrative clout to approve and implement new procedures if and when it is considered desirable to do so. The definition of our "user" becomes an individual who pays the bill, has sufficient management responsibility to insure that institutional policy is not changed without proper review and has sufficient clout to implement new procedures if such new procedures are justified by the new method of operation.

How much computing background does a user have to have? Clearly familiarity with computers and computing will not be a handicap to the user; however, the user does not necessarily have to have any computer background. Management of information handling may require a different perspective but does not require exotic or complex techniques. Time and time again, it has been demonstrated that simple and sensible management techniques are sufficient. Some of the perspectives important to information systems management are discussed in the following sections.

### The System Perspective

Information systems are "a complex interacting network of people, procedures and machines."[26] No amount of automation is going to completely remove the human part of the system equation. Thus, in dealing with systems we must deal both with the automated parts and the manual parts of the system. Subsequent to successful implementation, most system failures occur because the manual portion of the system was found to be operating below the level required for system success.

A system perspective also comes about by understanding the interdependencies amongst the various systems. A purchasing system must examine the accounting data base to see if funds are available before a purchase order is sent to the vendor and the purchase encumbered. The accounting system must depend upon the purchasing modules to provide the proper encumbrances so that the accounting statements are accurate. Similar relationships can be shown to exist between student records and housing, personnel and budgeting and position control and so on. If these associations and interrelationships did not exist, it would have been unnecessary to design them as integrated systems. An excellent visual demonstration of the interrelationships was presented in a recent issue of

----------------

[26] Jeffrey S. Lazarus, "On Executive Involvement in Administrative Computing," *CAUSE/EFFECT*, November 1981, inside back cover.

*CAUSE/EFFECT*.[27] Thus it is the responsibility of each administrative office to insure that there is substantial understanding of the big picture and their role and responsibility in it. Failure to do so will result in periodic finger pointing between the administrative departments, each accusing the other for its woes.

It is the responsibility of top management to insure that each of the department heads has a larger perspective, a perspective based on institutional priorities and needs which most likely are different from the local priorities of each department. This problem becomes particularly acute if a given department is considered the "owner" of a certain data item though it is not the only user of that particular data item. Unless top management clearly outlines its expectations of the various data owners and excercises the necessary authority to carry out these expectations, the system is doomed to failure. Integrated systems require a level of cooperation and coordination. If local politics are characterized by confrontations, the chances for a successful integrated system will be greatly reduced.

## Data Ownership

The manager who is closest to the origination point of the data item is generally considered the owner of the data. Ownership of data comes with an implied responsibility - the responsibility to insure that it is available in a timely fashion and that it is accurate. In most cases the owner of the data is also its heaviest user and has the best grasp of its significance. Because of integrated data base management systems, the routine access of all data increases tremendously. Greater use insures better quality of data. Greater access also brings to surface many of the discontinuities in information gathering and information handling. Hence it is impossible to clearly define ownership. The ownership of data will necessarily be dispersed throughout the organization; failure to define the responsibility for the data can only result in what will later appear to be lack of coordination.

---

[27] Jack Steingraber and Douglas Kunkel, "An Integrated University Online/Data Base System: A Reality," *CAUSE/EFFECT*, May 1982, pp. 12-13.

Control

Any system can be represented by a model consisting of input, process and output. The three stages of system sophistication, DPS, MIS, and DSS, were shown as a black box model in Figure 27. The input consists of the raw material used, the output defines the objective of the system and the process is some transformation that converts the input to the output. This model is very simple and very general. It can be applied to a component, to a subsystem or to the entire system. Models of subsystems used recursively define the model of the system.

In Chapter 4 we saw that the black box model can be applied to the three levels of systems we discussed. In a data processing system the input is data about a single entity, the output is information about a single entity and the process consists of selecting, sorting and otherwise handling records dealing with single entities. At the information system level, the input is information on entities, the output is integrated statistics and the process consists of aggregation along and across subsystems. In the case of decision support systems, the input is both internal and external statistics, the process is one of analysis and modeling and the output is decision alternatives.

Control systems vary only slightly from the black box model previously shown. Control is a three-part process:

- Setting of standards of performance
- Measuring deviations from the standards
- Correcting deviations.

In the control cycle the first responsibility of management is the setting of standards for the output of the system. It should clearly identify the expected outcomes. Next, a set of management reports must be designed to point out the deviations from the stated standards. The measurement of deviation varies from system to system and depends largely on what is perceived to be the output of the system. Two important methods of measuring deviation are through the use of error reports or exception reports. Correcting deviations is a management role. In fact, correcting deviations is management. A key element in the control process is feedback. Feedback is the recycling of information, new or old, so that inputs and processes may be adjusted to better, or more closely, meet the desired outputs or objectives.

In dynamic organizations, external influences like market conditions or government regulations might require that the outputs be changed from those previously defined. Or, even though the system may be operating as designed, new realities might require new outputs and hence modifications of the existing system. A corrective action would be needed through the control process.

As stated previously, deviations from expected results are, or should be, reported to management at the appropriate levels of the organization. Corrective action can only be taken if someone reads the reports, understands that a deviation has taken place and initiates the corrective action by proper feedback into the input process. If the reports are not read, the process can never be controlled.

In several cases it is possible to measure the deviation of the output and take some predetermined corrective action. These cases are candidates for automatic feedback control much as a home thermostat provides automatic feedback control. Automatically blocking a student from further registration when the GPA drops below a satisfactory level is automatic control. Dropping a student's classes for nonpayment of fees is automatic control. If, on the other hand, it is found that, say, 90% of the students have been dropped from their classes for nonpayment of fees, then perhaps there is a bug in the program or the process that completes this task.

Control is a function for all levels of management at all stages of the process. There are usually three types of controls:

- Preventive controls
- Alarms — or controls for detection
- Corrective controls.

The objective of preventive controls is to reduce the frequency of error and hence reduce the exposure due to these errors. The frequency of error is reduced by reducing the opportunity for error and by proper training of the staff. In the development stages of the information system, preventive controls come from project management techniques. The user's major responsibility is to insure that the system being designed will fulfill all the agreed upon needs. This is done by close and continuous interaction with the development team.

Preventive controls are easily applied to the input. This is done by edit checks on formats and ranges and by controlled values of data. There are certain errors that cannot be controlled by the system. For instance, assume that a data record contains the codes F and M for the sex of the student. Further suppose that the admissions office finds that they have no immediate need for the data on sex. However, they do have a pressing need to distinguish between students who apply for freshmen admission and those who apply for admissions to the masters program. Quite arbitrarily they decide to use F for freshmen and M for masters. The edit routines will not catch this change, as F and M are both valid codes for the system. Clearly the integrity of the data has been completely compromised for another user (say the housing office) unaware of this local change. It is therefore essential that each user use the system as designed. This is the most important preventive control. Changes to the system or to its intended use should go through normal change processes.

Preventive controls in the output stage relate to matters of security and exposures therefrom. In a batch mode where results, reports, checks and other sensitive documents are printed, it may require that special handling procedures be implemented and documented. Controlled distribution procedures should be used. In online systems, various security procedures can be built. Access to the data requires that proper DBMS conventions and procedures be followed. This knowledge, or lack thereof. forms the first level of security. The second level of security comes from the data submodel. The applicaton program or programmer knows about and works with only that subset of data that is needed to fulfill the job requirements. The third level of security comes about by controlling the terminal access to data. This type of security is at a function level. The Registrar's terminal for instance, can see student data but not payroll data. The fourth level of security is at the data level. Data level security comes in two forms: individual security and group security. It might be desirable to permit only one or two individuals in the payroll office to change salaries. For individual security the restriction is not on the value of the data but rather on the individual that wishes to access it. In the case of group security, the restriction is on the value of the data that might be displayed. For example it might be appropriate to allow all departments to look at accounting data. Terminals in the departments would accordingly be authorized to look at accounting data. But a further restriction might be that Mechanical Engineering can look only at Mechanical Engineering data and Electrical Engineering at

Electrical Engineering data and the Dean of Engineering can look at all Engineering data but not Arts and Sciences data. This type of restriction is on the value of data being displayed, not the type of data on display.

Alarms, or controls for detection, come in the form of error reports and exception reports. They also come from reviewing the contents of short life files like suspense files and clearing accounts. If the contents of suspense files continue to grow, it is a sure indication of trouble. If clearing accounts threaten not to clear, then we have a serious accounting problem. It is the manager's responsibility to look at these alarms and take corrective action.

Corrective controls are controls that are built into the system to help identify the problem, not correct it. In computer programs an error statement that specifies the nature of the error, where in the program the error occurred and when the error occurred is an example of a corrective control. Corrective controls anticipate the occurrence of the error and preplan the gathering of data for its resolution.

The most important corrective control is the audit trail. Audit trails are records that contain enough information to permit the reconstruction of past events. Audit trails provide the ability to follow a given transaction through the system at different points in time or to examine all transactions in the system at a given point in time. Most application systems maintain an audit trail of their own. This is most helpful for error correction or for detecting system misuse. In addition to the application audit trail, the data base management system has a system log that acts as an audit trail even though it is primarily designed for recovery (i.e. reconstruction) purposes.

In summary, then, a data base system should be secure. This security comes from its ability to prevent improper and unauthorized use; its ability to maintain a comprehensive audit trail; and its ability to recover from an inadvertent or intentional damage. These characteristics are possible if management exercises the proper controls and the users of the system are identifiable and the system monitors their activities through audit trails.

## Planning for Change

Unless software technology changes radically in the next few years, the development of information systems will continue to be a slow and expensive process. In order to reach the goals of the institutions over an extended period of time, a master plan for informaton systems should be developed. The master plan should be based on a desired operating and management environment. The aspirations for an operating environment in the future must be based on some perspective of what the future will look like. If the master plan is based on concepts, equipment and technology that exist today, then by the time the systems are implemented they will be out of date and relatively speaking inefficient. Because of the rapidly changing technology, the master plan has to be an anticipatory plan based on the economics of the future, not the conditions and requirements of the past. Further, the plan should be reviewed and revised periodically to reflect the changes in the environment. Since the planning horizon for such a plan is four to five years, it is important that the plan be an evolutionary plan that specifies not only where the institution needs to be in the future but also how the future components are going to co-exist with the existing systems and procedures. This plan will be extremely useful to the system designers and the users alike.

The next step in the planning process is to identify those areas that play a critical role in the success of the plan or have the most immediate benefits. These areas should be outlined for initial implementation. Most institutions proceed sequentially through the implementation plans, taking one system as a pilot and scheduling the other sequentially behind the first. This process has many hazards. Requirements of other systems are not easily available and therefore cannot be planned for in the pilot project. More importantly, however, if it seems appropriate to buy a software system (such as a student records system) then this procurement will proceed in isolation. A vendor supplied system that may be very re[illegible]nable from a single application viewpoint might be totally inadequate from a total systems viewpoint.

It is recommended that three or four areas of development be chosen as critical applications. The functional requirements of these areas should be completely defined. Armed with these functional requirements one may now proceed with the implementation of a pilot project in one of the areas. Should it be considered appropriate to acquire the software from outside, then at least the proper questions can be asked about interfaces with related systems and judgements made on the responses.

Thus far we have said that a master plan should be developed. This master plan should anticipate future technology and be subject to periodic reviews. A core group of functional areas should be identified for implementation. The functional requirements of these areas should be specified. Armed with this information we should proceed with the make versus-buy decision.

The decision to develop the software in-house or to buy it is an important one. There is no clearcut answer. The economics and politics of the local environment will be the major determinants in this decision. The alternatives to in house developmentare:

- Software exchange, either directly with another institution or through CAUSE[28]
- Joint development with another institution
- Purchase from a vendor.

All these alternatives should be examined. In-house development promises to be time consuming and expensive. Outside software generally cannot address all the requirements or peculiarities of the institution. Each institution is distinctive and should be. We do not need a monolithic education system. Whereas institutions are indeed different, some of the administrative processes are common. Thus there is a great potential for acquiring very satisfactory software form the outside. Regretably most, though not all, of the vendor supplied application software for institutions is based on outmoded technology and operating procedures. It would be fair to say that if someone gave away a completely batch student record or accounting system, it would not be worth taking, for its price in the long run would equal its value. This statement should not be construed as a complete denunciation of batch systems. Some applications work very well in the batch environment - like automobile registration or food service menu planning. They work better in the online environment but the benefits are not as dramatic as in the case of libraries, student records or accounting.

---

[28] CAUSE is the professional association for those engaged in the development, use and management of information systems in higher education.

In Chapter 4 we pointed out the deficiency in the present development process. The major deficiency is related to the level of involvement of the user in the entire process. The real question is to find an optimal mix of individuals in the project consisting of both users and computer professionals that will together achieve the project objectives in a reasonable timeframe at affordable costs. An attempt was made to model software manpower costs[29] associated with the development of software. The model took several variables into consideration including overtime costs, costs of interaction with users and others, administrative and other nondirect costs, average duration of work interruptions, average time to regain the train of thought after an interruption and so on. Whereas we do not necessarily agree with the methodology used in the paper, its conclusions are far more acceptable and agree with some of our own management experiences. The paper concludes that the cost of a project increases exponentially with the number of programmers assigned to it.[30] Indeed, one of the challenges of systems design and project management is to keep costs linear, both with the size of the project and with its complexity.

The paper also concludes that in all cases project costs are minimized by a project team of one. This conclusion says nothing about the completion time of the project. Our experience indicates that for complex projects a very small and very competent team gives the best overall results.

In light of the previous discussions a project team ld be organized as follows:

- One user representative assigned exclusively to the project
- One or two computer professionals assigned exclusively to the project
- One individual from data administration assigned only when needed to the project.

----------------

[29] See Bob Esterling, "Software Manpower Costs: A Model," *Datamation*, March 1980.

[30] The reader is referred to the now classical book by Frederick P. Brooks, Jr., *The Mythical Man-Month* (Reading, Mass.: Addison-Wesley, 1975).

The responsibility of the user representative is to act as spokesman for the user needs. It is the responsibility of this individual to talk to the various groups to determine their requirements and keep them informed about the direction and progress of the project. In a sense, this individual is the interface to the "outside world." Complete responsibility for functional specifications, user documentation and some portions of user training should be with this individual. Moreover, the individual should be very familiar with the area under consideration for automation.

The computer professionals should work to rapidly implement the essential operating components of the system leaving reports and bells and whistles for later. They should be working directly with the user without any written design or programming specifications.[11] Written material should be used only when it is deemed necessary to clarify a situation that verbal and pictorial (hand drawn) means could not. These individuals should also act as channels through which additional resources are funneled to the project on a short term *ad hoc* basis.

The data administration group should be involved during the data base design stages. They should also be involved if new data items are introduced into existing data bases and if the values and the uses of existing data items are to be changed. Considerations of security and authorized use of data are also items in which the data administration group should be involved with the development team. Finally, it should be the responsibility of data administration to see that the auditors are appropriately involved. The system should meet the auditability requirements. However, under no circumstances should the auditors be designing the system. They should be evaluating the proposed design for possible exposures and potential risks. The decision to take the risks is a decision of management: a decision properly made by the user and the developer.

Periodic project reviews should be held with the members of the users group and with other members of the system development group. These reviews should provide previews of the system whenever possible. The overall strategy is to reduce the cost and increase the

---

[11] It is only fair to point out that the opinion expressed herein is not universally subscribed to. For example, see Tobey L. Miller, "Applications Development: A Wholistic Approach," *CAUSE/EFFECT*, July 1982, p. 13, 25.

effectiveness of the development process by having a very small project team; to bring in all parties involved with the project or affected by it on an infrequent but regular basis to share with them the progress of the project; to tap additional resources only when needed; and to design and quickly implement a basic system.

Once the basic system is in operation a large number of people can be assigned to develop the reports and other features that a system always needs and without which it would not be a productive or useful system.

Once complete, the operational responsibility for the system must reshift to the user. The user should do everything that is required on a day-to-day basis to make the system work. This includes data entry, updates and the running of scheduled and *ad hoc* reports. The user office should have one or two computer professionals on their staff to help coordinate the operations of the system.

Organizationally then, the information system activities should be supported by three units:

- The user with complete operational responsibility and with responsibility for minor maintenance.
- The systems development staff with responsibilities for new systems development and major system modifications but with no day-to-day operational responsibility.
- The Computing Center which provides the raw compute power and acts like a computer utility.

In addition to these three groups there is a need for a fourth group called the Data Base Administration group. This latter group should be responsibile for all the data within the organization - for its availability, it security and its integrity. They should also act as technical consultants for the user groups that are managing their own systems. The individuals assigned to the user offices should be responsible for that portion of the data that they "own." The Data Base Administrator should have global responsibility for the entire data resource. Notice that the Data Base Administrator is a support group and at the same time a control group for the decentralized operations group.

User developed applications are written into the future of computing. It is the only way software can keep up with hardware. Hardware is moving too rapidly for present software techniques and practices to remain viable. The decentralization of the operational responsibility for administrative systems is an important step in preparing for the future.

## Software Directions

Most of the languages in existence today, at least those languages that are used with data base management systems, are procedure oriented languages in that they specify step by painful step what the computer needs to do. It is the rigor of this procedure that makes writing software so difficult and expensive. There is a newer generation of languages used with data base systems called "nonprocedural" languages or problem-oriented languages. These languages do have procedure but only in a macro or broad sense. They lack the detail level of specificity required in languages like FORTRAN and COBOL.

The nonprocedural languages, also called fourth generation languages, are getting popular. Examples of some of the fourth generation languages are FOCUS, NOMAD, RAMIS and INQUIRE. We have already said that the direction for Data Base Management Systems is the Relational Model. Its conceptual simplicity and ease of use can easily make it a powerful tool in the hands of the computer professional and the user. A combination of a powerful fourth generation language with relational data base will go a long way towards addressing some of our software problems.

The literature reports truly dramatic performance improvements that are possible by the use of these languages. The Bank of America has 500 NOMAD applications running, all of them user developed with the help of a few consultants.[32] FOCUS, too, has shown that much improvement is possible with fourth generation languages. FOCUS has a language that contains a wide range of English verbs and English syntax.[33] FOCUS also supports an "integrated" relational data base. Integrated is defined to mean "that every single relation in the data bank can communicate with every other relation via any like key field or data field."[34]

----------------

[32] Daniel D. McCracken, "Software in the 80's: Perils and Promises," *ComputerWorld*, 17 September 1980.

[33] We are not sure if that is a good idea.

[34] N. S. Read and D. L. Harmon, "Assuring MIS success," *Datamation*, February 1981.

It is designed to be used directly by the user. It has an integrated graphics package that allows results to be plotted with the same ease as printing them.

The fourth generation languages are inefficient in their use of computer resources. They may take anywhere between 50% to 500% more resources depending on how poorly the request and the supporting data match up. But with computer costs dropping this should not be a concern for too long.

Certain character manipulations are not easily performed in these languages. Applications that require this feature will have to depend on the COBOL, PL/1 or Fortran. The most important feature of these newer languages is that they permit "interactive development" or prototyping. A user spells out a requirement and can see it in operation in a couple of days or less and decide if it meets the requirement. If it does not, it is simple to go through another iteration and another till a satisfactory result is obtained. Once a satisfactory result is available the question [illegible] can be addressed. Should the task be [illegible] in a more efficient language? Or will the potential performance improvements to be gained not justify the [illegible]?

The experience with these tools is, as yet, very limited. The initial results are dramatic. We might soon have to take the bold step of divorcing ourselves from our existing software base in order to adopt the new technology. A big step indeed. But then one cannot [illegible] a chasm with [illegible] small steps.

## INTRA-INSTITUTION COMMUNICATION

"... That society can only be understood through the messages and communications facilities which belong to it."
Norbert Weiner

Communications is an important aspect of institutional administration. The ultimate product of an institution of higher education is knowledge knowledge that it creates and knowledge that it imparts to others. The written and the spoken word remain as the primary means of communicating this knowledge. The cost of creating, storing, retrieving, transmitting and reading the words and text generated on campus is reaching frightening levels. Stanford University reports a $35 million a year cost in text processing and communication alone. This does not include the cost of the author's time to create the material and the reader's time to digest it. The cost of photocopying alone at Virginia Tech exceeds $1 million a year. Nationwide, the cost figures on information transfer are even more staggering: $129 billion are spent annually on communications of which $4.4 billion are for typing alone.

What has this to do with university data processing? Everything! The issue is one of productivity and of effectiveness. A student recruitment program that can send out high quality personalized letters to prospective students stands a better chance at recruitment than one that mails out an uppercase only computer printed letter or, worse, a form that checks the applicable options. A fund-raising program would be relatively ineffective without "personalized" mailings. There are also similar examples within the institution. The faculty spend numerous hours proofreading the work of their secretaries. During each iteration in the document preparation process they have to repeatedly choose between making changes to improve the document or not making changes to avoid additional retyping for the secretary. Secretaries spend numerous hours retyping the various versions of the same document till it is finally mailed out. These are only a few examples of the tedious and the unnecessary and duplicate work that now goes on in our institutions. Experience indicates that automated text handling systems have an important place in our work environment. Where used in an effective manner they reduce the time for document preparation, reduce the work associated with it, and produce a better quality document. All this has to result in better productivity.

Are these automated systems cost-effective? There are two issues involved here. The first issue is one of economics. The cost of an average letter is more than six dollars today, up from a little under three dollars less than five years ago. Unless we can dramatically change the way we produce our mail, the costs are sure to continue to escalate because they are tied directly to personnel costs. A significant improvement in productivity is the only hope in containing these rising costs. The second issue is an issue of necessity. Most institutions are facing hard times. Dollars are limited and positions even more so. Improvements in the quality of our programs will come only if we make more productive use of all our human resources. This applies equally to faculty, professionals and clerical staff. The output of an institution is measured by the output of its faculty. At present, based on a very small sample, the personnel distribution in an institution is about one third faculty, one third professional and one third clerical staff. The application of text processing tools has the potential of either changing the ratio of this distribution or increasing the overall faculty output and productivity keeping the same ratio.

The growth in farm productivity has been a direct result of technology. An average farm worker is backed by $35,000 in capital equipment and machinery. An average factory worker has an investment varying between $25,000 and $40,000 (depending upon the industry) backing his work. In the case of the white collar worker, the average capital investment is reported as being only $2,000.

Raymond Kenshan argues that if an investment is made for capital equipment needed in the office environment, this investment will produce a greater return than the same investment in the farm or in the factory.[15] A $5,000 investment will increase the capital investment for a factory worker by 20%, but this 20% additional investment is not likely to result in a 20% increase in productivity. On the other hand a $5,000 investment in an office environment will not only double the present investment but will also change the entire approach to the work done in the office. We will change not only how we do things but also what we do. This can be truly significant as large numbers of people are employed in the office environment. Already 50% of the GNP is generated by the white collar labor force. In excess of 13% of the total labor force is white collar. Clearly large potential for significant improvement exists. The percentages are even more attractive in the academic world. Their product words certainly lends itself to this technology. Planning for office automation may indeed be the most crucial area in institutional planning.

The remainder of this chapter examines the various aspects of word processing and office automation and the physical means of transmitting the text and messages generated across the institution.

## Office Systems in Perspective

The functions performed in an office environment may be grouped into five broad catagories:

- Text processing
- Text distribution
- Text reproduction
- Records management
- Administrative support.

---

[15] Raymond Kenshan, President, Management Assistance Inc., New York, N.Y.

A text system will be construed to consist of the first two categories: text processing and text distribution.[16] Text processing is the function of reading, writing, revising, filing and retrieving text. In order for a system to be useful it has to be both available and easy to use. Availability implies that the computers supporting the process be up and that enough terminals be available so that a user has access to the computer system. The writing of text involves the entry of new text, merging of new text with old text or merging of old text with old text to form a new text. The ability to write in mathematical notations or foreign languages implies the existence of special fonts and character sets. The ease of revising of the text depends largely on the editing capabilities of the system. The ease with which a given line, page or word can be found determines the power of the editor. Once found, the ease with which new text can be added and old altered determines the productivity of the user. The reading of text is facilitated by justification, hyphenation, spacing, etc. of the text. The entire process is helped by automatic aids like checking spelling, numbering tables and figures, and automatic footnoting, indexing and generating a table of contents. Filing of text has two parts: ease of retrieval and security. Thus a filing system should be easy to use for the authorized user and difficult to use for an unauthorized user.

Text distribution may be accomplished in two ways: online and offline distribution. Offline distribution deals with copying, printing and mailing. It requires computer support for maintaining mailing lists and producing mailing labels. These are functions that we are all too familiar with. Online distribution deserves some careful analysis. There are two types of systems dealing with online distribution: electronic mail systems and electronic message systems.

Electronic mail systems are point-to-point communication systems. They transmit information from a source to a destination where it is printed. At present electronic mail is a $2 billion industry dominated by Western Union and ITT. There are two basic types of systems for electronic mail: teletypewriter systems and facsimile systems. Teletypewriter systems can only transmit characters of information. Facsimile systems, on the other hand, can transmit graphics information: thus letterheads, signatures and

---------------

[16] Parts of the next two sections have appeared previously in various consulting reports prepared by the authors.

figures can be transmitted along with the text. Naturally, facsimile systems require much larger bandwidths and therefore are more expensive to use.

Electronic message systems deal with all aspects of text processing. In addition to distributing the message, they are capable of providing automatic header information, time stamps, distribution lists and acknowledgements when the message is received and read; of redirecting the messages after additions or alterations; and of browsing, including filing and refiling. The size of the text in the message is an important factor in determining the kind of system required to support the activity. If the size of the text is small, it is called a *message*; large texts are called *files*.

Since the messages are small, the computer configuration supporting a message distribution facility is not as critical as in a file distribution facility. Online file distribution can be a problem creating large bottlenecks in the network unless adequate plans are made for it. The problem is related to the amount of data to be transferred and the capacity of the network to support the data. Let us analyse the problem.

There are three components in a text system: the computer, the storage disk and the terminal. Examine for a moment the data transfer amongst these components. The human interface to the system is through the terminal. Compared to the computer's ability to transfer data, the ability of the human to read is very limited. The data transfer requirements are constrained by the reader and hence pose no significant load on the system. The link between the computer and the terminal does not have to be a very high speed link; up to 2400 baud for character information and up to 19,200 baud for graphics data seem to be quite sufficient. Half these speeds are also adequate. Thus the link between the computer and the terminal poses no problems at all. The transfer of data between computer and disk is a high speed transfer. Disk drives are typically located near the computer and hence pose no new or significant problems for high speed transfer between them. Now consider transferring a file from one user to another. This implies that the data will be transferred from the disk of the sender to the disk of the recipient. This disk-to-disk transfer will take place at high speed if either both the users have their disks (not necessarily their terminals) located at close proximity to each other (i.e., centralized) or if the network that connects the two disks allows for high speed data transmission. Unless the files are centralized, file transfer assumes the existence of a high speed data transmission network. This is perhaps the most serious disadvantage of distributed word processing equipment it places too great a burden on the network if frequent file transfer is desired. Since we are moving more and more towards a file transfer environment, network capacity limitations should be a prime consideration in structuring the office automation environment. The requirement would favor centralized word processing.

There are several commercially available electronic message systems. Examples are Tymnet's Ontyme, Telenet's Telemail and Datapac's Envoy100. Message systems are also proposed in the new generation of commercial networks: Satellite Business Systems (SBS), Xerox Telecommunications Network (Xten) and ATT's ACS. The development of message systems have been quite rapid and, as is generally the case in any new product area, there are no standards. Several standards are needed if message systems are to gain in popularity and use. First, there are no standards for message structure. The problem of standards in this area is complicated by the potential use of multi-media messages, i.e., mixing text with voice, video and graphics. Second, there is no standard way of handling the undelivered message. An undelivered telephone call (busy signal) or an undelivered letter gives an indication to the sender, but this is not true, at present, in the case of an undelivered message. Third, unlike the telephone directory, there is no universal directory for identifying the individuals to whom the message is sent.

Even though there are no standards today, standards work is underway. In fact, there are several groups working on it: the teletex group of CCITT (Comite Consultatif Internationale), the working Group 6.5 of IFIPS (Intenational Federation of Information Processing Societies) and ANSI X4A12 working group 4 are all addressing this subject matter.

Reproduction concerns itself with the copying and printing function. In keeping with the philosophy of source point data capture, many institutions are acquiring typesetting equipment that is driven either online or offline from their word processing operations. This connection eliminates the work, delays and expenses associated with manual typesetting. There is also a trend to use the high speed printers connected to computers as copiers. It is easier to have the line printer or page printer print the required number of copies rather than producing one copy from the computer and xeroxing the rest. Budgetary considerations and charging algorithms might, however, be the final determinant of how the document is produced.

The whole area of administrative data processing addresses itself to records management. Records management includes both computer records and paper records (like letters of recommendations, purchase orders, etc.). It also includes the permanent archiving of the computer generated records. The use of the computer opens the possibility for automatic archival and retrieval of archived records.

Administrative support deals with things like calendars (schedules), facility scheduling (conference rooms), travel reservations, reminders and related office chores. Whereas all of this can be done with the help of a computerized system, there is a matter of privacy and control that must be addressed. By placing the calendar on a computer system one gives up the liberty of determining which meetings not to schedule; i.e., it allows someone else to control the schedule. Since this unknown assault on the calendar changes social behavior, this particular application is not likely to be very popular. The privacy question, of course, deals with the fact that the calendar is available for others to see. Whereas schemes can be devised to limit the exposures by using complex pre-empting algorithms to determine who may or may not see your calendar, it is not clear that the function will become popular anytime soon. However, these functions will indeed be indispensable if and when the white collar labor force becomes a cottage industry. Chapter 9 describes an interesting scenario associated with this possible migration.

In this section we have tried to outline the various functions that would be necessary in a properly automated office. The key point to note is that office automation is more than word processing. We also made a distinction between electronic mail and electronic message systems. The rest of this chapter will deal with different aspects of word processing and with the facilitating network to transfer the data around campus.

## Word Processing Attitudes

Virginia Tech recently conducted a study on the nature and use of word processing on campus.[37] The study was directed at department heads and secretaries and was

---

[37] This section is adopted from Virginia Tech Systems Development Technical Report Number 80-004-001, May 15, 1980.

conducted by a structured interview process in which the department head was interviewed. This was followed by an interview with the departmental secretary.

The study pointed out some very interesting facts. Nearly 35% of the department heads indicated that their secretaries knew how to use the centralized word processing facility (CMS SCRIPT), yet only 17.4% of the secretaries indicated that they had ever used the facility. This clearly points out that many department heads like to think that they are using all the available tools, though in reality, they may not be. Several disciplines in an institution do not have any need for direct or obvious application of computers. It is generally in these disciplines that the chief administrator has no exposure to computers or to computing. They depend entirely upon their assistants or secretaries to use the data processing capabilities. It is, therefore, not surprising that equipment procurement requests coming from these departments are based largely on the experiences of the secretaries or administrative assistants in the departments. It is, therefore, in the interest of the institution to expose the department heads to the capabilities of the various alternatives in word processing and outline the tradeoffs involved in choosing one system over the others.

The department heads were also asked if they would prefer their own stand-alone system rather than using the institution's central facility. Over 87% indicated a preference to using their own equipment because the central facility was "unwieldy and unreliable" for their purposes. The remaining 13% preferred the central facility because of "downtime" of their own equipment. This observation must be viewed in light of the fact that only 47% of the departments were even aware of the existence of centralized facilities, 35% thought their secretaries used it, 17% of the secretaries actually used it and only 9% of the department heads had first hand experience with word processing. Further analysis of the results of the survey showed that, by and large, the department heads that did not have stand-alone systems preferred stand-alone systems because "they would be more reliable" and those that did have stand-alone systems preferred the centralized system because it "would be more reliable." Clearly good service and not equipment location is the chief factor influencing preference.

Various output samples were also presented to department heads. They were asked to choose the samples they liked best. From the samples chosen, the results suggest that the type of paper is a more important determinant of print preference than the type of printer.

A department head was willing to spend on the average $6,000 on WP capabilities and up to a maximum of $30,000. The capabilities that were most frequently mentioned, in order of importance were as follows:

- Full screen text editor
- Capability to handle special symbols
- Mailing lists merged with letters
- High quality fast printer
- More memory
- Diablo printer terminal
- Mailing lists for postcards.

Amongst the recommendations of the Virginia Tech study was one that said that stand-alone or shared logic WP systems should be acquired only after it has been demonstrated that centralized facilities were used and found inadequate. This recommendation is particularly interesting in light of the fact that 87% of the department heads indicated a personal preference to acquire their own equipment.

A study at Stanford arrived at similar conclusions. An excellent report on text processing was published by the Committee on Office Systems and Technology (COST) at Stanford University. The report is entitled, "A Network-based Text System for Stanford." According to the report, Stanford consumes three tons of paper a day. It spends an estimated $35 million in preparing and communicating messages and text. About $2 million a year are spent on the acquisition of stand-alone word processing equipment.

Stanford's study indicated that none of the presently available commercial systems would meet their defined needs. They also concluded that building their own custom-made system or having a system built to their specifications was not a feasible alternative. As a result, their present recommendation is to expand the campus-wide centralized system and, using it as a base, provide a set of integrated services to their faculty, staff *and students*. They also recommend that the number of different vendors for stand-alone systems be limited to a very few. A diverety of equipment for distributed systems where needed would only make the cost of integration prohibitive. They are, however, discouraging the procurement of stand-alone systems and encouraging people to use the centralized facility. They believe that the revolution in our work environment will come not from using specialized or sophisticated equipment but rather from the large body of individuals using available services and facilities on a regular and routine basis. Amongst the stated objectives of the Stanford report is to disseminate their report and seek institutions and businesses that might work cooperatively with Stanford to further the objectives outlined in their proposal.

In recognition of the present state of the art in word processing technology and the stated requirements for connectivity amongst the various users in an institutional setting, we recommend that with a few exceptions, institutions should invest in centralized and not stand-alone word processing capabilities for the following reasons. First, the offerings available in stand-alone word processing equipment do not address the total office automation concept described above. Second, personal computers are becoming extremely popular. We believe that the enhanced capabilities of the personal computers - including text processing - will lead these computers and not word processors to be an integral part of the office work station of the future. We are likely to see machines that exclusively address word processing become extinct. The more general and versatile personal computers will indeed replace them. Third, stand-alone systems mean that word processing would only be available to departments able to afford/acquire the equipment; on the other hand a centralized system would be available to all faculty, administration, staff *and students*. Fourth, a centralized system would be in less danger of obsolescence as it is supported primarily by software. Fifth, any upgrades to the system would be simultaneously available to all users.

The strategy of the computing center should be to provide a very high quality of service to the user community. This service should be the standard against which the offerings of stand-alone and other systems should be judged. This requirement implies that the centralized system would provide full screen editors that drive high quality printers that can print on cut sheet stationery and even use letterhead stationery. Further, an interface to the printing plant would be desirable. Direct terminal to terminal connections amongst users is considered essential. Capabilities to invoke a variety of functions like mail, message, data, calendar, reminders, spread sheets, etc. would help make the systems more valuable to the user community. The need for possible local printing should not be ignored.

It is, indeed, very likely that the centralized system can not economically address all requirements of all users. In case of clearly demonstrated requirements that are unavailable on the centralized system, like the need for some very special fonts or character sets or the necessity for extreme confidentiality — as in the case of the President's office — a stand alone system may indeed be very justifiable.

An easy to use and powerful centralized word processing system with high quality print capability and a connecting network for communications is clearly the recommended strategy at this point in time. The network that can provide the universal connectivity will be the subject matter of the next section.

## Local Networks

Computer and communications are two rapidly merging technologies. The distinction between them is all but erased. The recent antitrust rulings open the doors for some very exciting opportunities in this field. The introduction of satellite communications has made it very simple to send information clear across the country. However, the difficulty still lies in distributing the information locally when it is received. It is no exaggeration to say that it is easier to make data flow from San Francisco to New York using satellites than it is to get it to flow from First Avenue to Fifth Avenue in New York. A new technology called local area networking is emerging as a solution to this dilemma. The technology has immediate application in our institutions.

Institutions of higher education are, at present, likely to have at least three different networks in operation. The most common network is the telephone network. Several institutions have a cable television network to broadcast educational materials to the [illegible]rooms and the conference rooms. Institutions with [illegible]mputing capacity have a network that supports ter[illegible] remote job entry stations and computer-to-computer [illegible]ications. In addition to these three networks some institutions have a security network and others have or are working towards a network for energy management in its facilities. This section deals primarily, though not exclusively, with communication networks that may be used to address all the above application areas.

We will, however, concentrate on local networks for the support of computing.

There are several considerations that lead to a need for a network. First, a terminal is typically not in constant use and hence there is no need to provide a computer port for each terminal. A single port can generally support between two and three terminals. This makes it advantageous to provide some port contending capabilities on the computer systems. Second, it makes exceedingly good sense to create an environment in which any terminal can access any of the machines located in the computing center, or better still, located anywhere on the campus. This strategy makes good use of the terminals and precludes the need for an individual doing a variety of computing on different machines to have more than one terminal.

In the past, port contention and generalized access have been provided through the use of dial-up ports on the computers and acoustic couplers on the terminals. The advantage of this approach was that the existing telephone network could be tapped for this purpose. Dial-up support, though very useful for very remote access, is generally inadequate for terminals located on campus. The sign-on procedure for dial-up terminals is cumbersome at best. The terminals generally run at low speeds of 300 baud to 1200 baud. Transmission at higher speed over voice grade lines is neither cost effective nor reliable. Further, acoustically coupled modems running at higher speeds tend to cost more than the terminal itself. Most users like to run their terminals at higher speeds due to the efficiency that results from it; some users - particularly those using graphics - have to run their terminals at higher speeds in order to get any work done. Thus, along with port contention and generalized access, higher speed transmission becomes the third requirement for computer networks.

Two widely differing techniques may be used for addressing the requirements. The first method employs a data switch to provide the capabilities of automatic speed detect, port contention and access to multiple machines. A communication system using the data switch requires the data switch itself, a modem at each end, and a telephone circuit which may either be a leased line or a switched line. The telephone line begins at the terminal and terminates at the data switch. The data switch is connected to the various ports on the computers and connects the terminal to the desired service if it is available.

The use of the data switch is a reasonably effective method of providing the services of [illegible] well as long as the aggregate data transfer rate is not much larger than [illegible] represents approximately a thousand terminals running at speeds up to 2400 baud. The capacity is low for devices requiring higher speeds. With the growth of micro computers and the increased use of graphics terminals, the need for higher data transfer rates will be inevitable. Further, the data switch still represents a star network and each new terminal installation requires the installation of a new telephone line. The data switch, though a very useful option for the present, cannot be [illegible] to provide for computer communication needs [illegible] the decade.

The other option for addressing the data communication needs is the possibility of using a local area network. A local network uses a medium that provides a wide bandwidth (aggregate data carrying capacity) and, hence, permits the simultaneous connection of several thousand devices. The communications medium may be a coaxial cable, a fiber optics cable or radio broadcasting. Local area networks use one of two techniques to allow many devices to share a common transmission medium. These two technologies are TDMA (for time division multiple access) and FDM (for frequency division multiplexing).

In TDMA networks each node of the network may access the transmission medium only at certain times governed by the network interface. The network interface has to define some scheme that permits the supervised and orderly access to the transmission medium. One of three techniques is generally used to supervise access to the medium: polling, reservation or contention. In *polling schemes* the network controller is constantly checking the network stations for requests and then allocates channel times to the requests. *Reservation schemes* depend upon the nodes to access the communications medium only at predefined schedules. *Contention schemes* require the nodes on the network to compete for the network medium. Ring networks operating under the contention scheme use either the slotted ring or the token passing scheme for competing for the medium. Bus networks, on the other hand, use a technique called CSMA/CD (for carrier sense multiple access with collision detect) to manage contention on the transmission medium.

The whole point of the above description is to point out that networks that operate under the TDMA scheme allow only one user on the network at a time. Different [illegible] the network at different times. Signals that are pumped into the network medium do not have to be modulated in any way. Networks using this technique are called baseband networks. Baseband networks using coaxial cable as a medium can transmit data at ten million bits per second over a distance of a mile. The effective data transfer rate, of course, is much lower.

Local networks may also share the transmission medium using frequency division multiplexing (FDM). FDM is identical to radio broadcasting in which the data signal is superimposed on some carrier signal. Different nodes on the network [illegible] at different frequencies. Thus many nodes [illegible] share the network at the same time. Broadband networks use this mode of operation.

Coaxial cables can be used in both broadband and baseband mode. The cable used for baseband mode differs only slightly from the one used for broadband transmission. The former cable has a central carrier wire surrounded by a braided copper, while the latter consists of a carrier wire surrounded by an aluminum shield. The transmission rate for broadband cable is 150 million bits per second and represents a substantially greater capacity than the baseband mode.[…]

The use of broadband coaxial cable is certainly not a new technology. Coaxial cables have been used for several years in cable television networks. The great potential of using broadband coaxial cables in local networks is that it provides for the communication of telephone, video and data transmission over a single cable. This is possible due to its wide bandwidth. Transmission is done on channels defined by frequency division multiplexing. Some channels can be reserved for TV transmission and others for voice and data transmission. Over 10,000 terminals at 9600 baud can be supported on a single cable. The number is even greater at slower speeds.

As stated earlier, there are three possible media that may be used for local networking. Of the three, coaxial cable and fiber optics are the most suited for transmission on a campus. Fiber optic cables have extremely high bandwidths and can operate at data rates up to 200 million bits per second. Because the mode of transmission is o

---

[…] See M. A. Dineson and J. J. Picazo, "Broadband Technology Magnifies Local Networking Capability," *Data Communications*, February 1980.

is insensitive to electromagnetic interference and noise. Since no electrical signals are passed through the cable it is free from illegal wire tapping, thus providing much greater security. The major drawback of fiber optical cables is that it is very difficult to tap the signal. Cables are essentially custom made to prespecified lengths. So far their use has been limited to high speed point-to-point links.

Most local network designs are based on coaxial cable networks. Xerox Corporation's Ethernet was developed in 1976 and uses the baseband mode of transmission. Network devices based on coaxial cables operating in a broadband mode are available, amongst others, from Amdax Corporation and Network Resources Corporation.

Ethernet is a serial-bus local network that uses fixed length packets that contain data and header information identifying the destination location. The network can have up to 32 processors each with a network interface module that formats and transmits the data on the network. Ethernet is a contention network using the CSMA/CD scheme. There is no central computer managing the traffic on the network. Each node monitors the signal on the network and transmits data only when it finds a free time period. If two nodes start to transmit at the same time they both stop for a random interval of time. Retransmission is begun after that interval if no one else is on the network.

Hyperchannel is also a computer-to-computer network based on the coaxial cable that can be multidropped at up to 64 locations along a mile of cable. Contention on the hyperchannel is handled differently from the Ethernet technique. In hyperchannel data is transmitted in frames that vary in size (up to 4K bytes). Each transmitted frame requires a response frame from the receiver. If two nodes transmit at the same time, there is no response frame generated due to the error in transmission. Detecting no response the node with a higher priority retransmits the data. The lower priority node waits its turn.

The LocalNet network offered by Network Resources Corporation (Sytek, Inc.) is based on broadband mode of transmission on coaxial cable.[12] It supports video, voice and data communications at varying speeds on a single coaxial cable in a 25-mile radius. Digital data is

---

[12] See LOCALNET, Network Resources Corporation, Mountain View, CA for technical details.

transmitted using packet switching techniques. The bandwidth of the coaxial cable varies between 300 and 440 MHz. The system uses 120 contiguous 300 KHz channels for its network operations. All units connected to the network transmit at the lowband frequencies and receive at the highband frequencies. The transmitted data first goes to a converter or translator unit where the lowband is converted to highband frequencies and retransmitted. This arrangement is called a mid-split. The retransmitted data is captured by the appropriate section of the network intelligence and sent to the connected device. The ease of system definition and its great flexibility are major advantages to this approach of networking. This networking scheme can most easily be integrated with cable TV networks providing a new dimension in connectivity.

Brown University has recently completed a study on local networking.[40] The study recommends that the University should implement a local network based on the broadband coaxial cable technology Their design consists of a redundant cable system for greater reliability. They propose that two identical trunk cables be used to connect the facilities on campus. One cable would carry the "core services," and the other would carry "special or scheduled services." Examples of special services were high resolution graphics and high speed computer to computer connections. Both cables would carry network control, maintenance and security services. The idea is that failure in the normal services cable could be recovered from by temporarily using the other cable for normal services. The special services would be interrupted till such time as the cable remained unavailable.

The applications of the network would consist of the following:

- Security and safety
- Auxiliary fire and smoke alarms
- Intrusion alarms
- Remote TV surveillance
- Building access control

---

[40] W. Shipp and H. Webber, "Final Report: Study Group on Telecommunications and Networks," Brown University, 1980.

- Alarm status monitoring
- Emergency broadcast system
- Medical emergency alarm system
- Energy Management
- Data Communications
- Education and Entertainment TV
- R. I. Health Network
- Community Services.

Local area networks are going to play an increasingly important role in our campus communication systems. The use of broadband networks for voice, video and data communications may indeed be the norm in most institutions by the mid eighties.

## Terminals for Managers Program

The Stanford Terminals for Managers Program was initially conceived as a one year experiment. The program provided a terminal and supporting software tools to selected administrators, with the primary mission of introducing new technology to the senior and executive officers of the university.[41] Initially, 55 terminals were located in the work areas of the individuals involved in the program. Text processing and electronic message system were the functions provided, along with a filing system and a reminder system. The standard features of each of these four functions were available to all users. In addition, a directory was provided which translated three character name abbreviations (initials) into computer accounts. A feature for resolving duplications was built into the directory function. The directory also allowed the identification of a commonly used subgroup of individuals receiving messages.

---

[41] See Cedric S. Bennett, "Stanford University's Terminals for Managers Program," *CAUSE/EFFECT*, May 1981, pp. 22-25.

The project team provided one-to-one education to the managers in the program. Several levels of documentation were provided along with an online help feature. In order to make the users more comfortable with the system, a variety of game playing software — StarTrek, Adventure, Solitaire, BlackJack, etc. — was made available. The participation in game playing dropped off as work related activities increased on the system. The goals of the experiment were to improve the communication process, to provide the personal use of computers for problem solving, to support a variety of office arrangements and to create a friendly and easy-to-use system that would sell itself. Of the four goals listed above, the personal use of computers for problem solving was not implemented initially due both to the diversity of problems and the requirements for solution tools.

The experiment has met with tremendous success. From the initial group of 55, the number of users with terminals that are participating in the program is up to 350. Over 250 of these users pay their own bills for the use of the system. When the system was first implemented there were an average of four messages per person per day. The number has since grown to ten messages per person per day, testifying to the acceptance and success of the program.

Motivated by the success of this experiment, the project team is planning several enhancements to the present offering. The software is scheduled for change to improve its efficiency. The decision support tools that were excluded from the first experiment are under consideration for installation. Access to specialized data bases like the Capital Budget file and the Legislative Bills file is now available.

The Stanford experiment is significant in that it refutes the widely held belief that executives and managers will never use the terminal. The key element for the success of the program seems to be the fact that a very simple, easy to use, convenient and useful application was provided to the managers. Its apparent success has bridged the fear gap that many individuals have when it comes to computer systems. The real challenge will be to continue to build on the base of knowledge and the behavior pattern that this experiment has generated.

## Planning for the Automated Office

The automated office concept is still considered by some as "futurist." The rate at which technology has been changing and the continuing blurring of the distinction between computing and communication points inevitably to a changing work environment. Office automation is one project that cannot even have a beginning without top management support. Because a main component of office automation is electronic message systems, a critical mass of users is necessary to begin the program. It also needs a well thought out facilitating network. Both these requirements call for front end financial support without which the project cannot get off the ground.

Once top management agreement and support is available, the next step is to put together a task force that consists of individuals involved in computing, administrative services, personnel management, behavorial sciences and space planning. Because a properly implemented office automation project has the potential of radically changing the work environment and how we interact with it, the participation of the various members in the task force is crucial.

Who should this task force report to? Is this a function that is properly located in computing or in administrative services or somewhere else? It does not really matter. The task force should report to the university administrator that has the best business sense and believes in the concept.

A strategic plan then needs to be developed that considers several aspects of the process. First, a conceptual plan is needed that describes the scenario ultimately to be achieved. This will help influence the day-to-day decisions that will lead to the stated objective. Second, an organizational approach must be agreed upon. Should the machine intelligence and processing capability be centralized or decentralized? Or should a combination of the two be used? Local realities, the size of the organization and its present environment will define the best approach to be taken. Third, the functionality of the system should be defined. Sophisticated system functions can be planned for a staged implementation as the user experience, understanding and acceptance grows. Thought must be given to ease of use and ease of learning characteristics. Richness of function must be weighed against increased complexity of the system. The design should encourage the frequency of use. Fourth, the hardware approach should be defined. It is important to standardize on one or two (no more than three) different types of equipment. This standardization will tend to perserve the investment in training time and allow the user to use the system from multiple locations with the same effectiveness. Fifth, an evolutionary implementation plan must be developed. It is important that members of the task force (and some of their friendly peers) along with the members of the computing center should make up a pilot group to debug the hardware and the software. Next a large enough group from the university community should be chosen for initial implementation. Interfaces to normal communication media should be provided so that the user does not have to duplicate the work. For example, assume a user wishes to send a message to a group of eight individuals. Only five of the eight have terminals on their desks. The user should be able to send the message once five individuals would receive it on their terminals, the other three messages would be printed in the mailroom for hand delivery.

The social impact of this technology is likely to be considerable. As a general rule, people do not resist technology, they resist social change. The person-to-person communication will no longer be the same. The lack of body language and other nonverbal cues in electronic systems will somehow have to be compensated for.

Perhaps there will be an increased use of explicit verbal cues in our language patterns. Or multimedia communications - using voice with text and graphics - will become the norm. The secretary/boss relationship will definitely change it is not clear how. Because of these changes that are likely and other changes that we cannot predefine, the participation of the behavorial scientists in the task force for office automation is very important.

## NETWORKS AND SHARING

"So, Nat'ralists observe, a Flea Hath smaller Fleas that on him prey. And these have smaller Fleas to bite 'em. And So proceed ad infinitum."
Jonathan Swift

### The Economics of Computing

To the user the most important impact of the computer system becomes the time it takes, from data entry to finished report circulation, to get his job done.[42] To management the most important consideration is that the work be processed in order of value to the institution (and as efficiently as therein possible). These are two conflicting criteria and each group must be sensitive to the concern of the other if the two concerns are to be brought into some commodious arrangement. This suggests that the institution might well establish an internal/external marketplace for data processing services. A marketplace analogy would also suggest the desirability of decentralization and networking as competitive tools.

---------------

[42] Material in this and the following section has been adopted from R. C. Heterick, editor, *Virginia Information Systems Exchange Network Plan*, November 1974. We are particularly indebted to Charles Goetz for his insights on the economics of computing.

Data processing facilities, like any other economic entities, should be subject to the fundamental economic question: how can these scarce resources be put to more efficient use? The advantages of networking become more apparent when the notion of "efficient" resource use is conceptually subdivided into allocative and technological efficiency. Of these, technological efficiency is the most straightforward, merely requiring that the same output mix cannot be produced with fewer inputs; whatever is produced must be produced in a technologically resource-minimizing fashion. Allocative efficiency asks a more thorny question: which particular output mix should be chosen, given that an almost limitless number of output constellations are technically consistent with any fixed resource endowment? In sum, efficiency is desired both as to how computer services are produced and also as to exactly which services will be produced.

Centralized management is in the best position to guarantee allocative efficiency when the conditions for cost-effectiveness analysis are satisfied in their pristine form. Specifically, the "effectiveness" of alternative output mixes must be susceptible to quantification. However, data processing serves a wide spectrum of extremely disparate uses which are relatively difficult to compare in a meaningful quantitative fashion. This non-comparability emerges for at least two major reasons. The first, and most important, is that the "effectiveness" of the uses will frequently be subject to widely divergent, although perfectly defensible, subjective evaluations, even by "expert" observers. The other is that, subjective factors apart, the very diversity of the data processing applications makes it unlikely that any centralized evaluating agency can realistically amass and coordinate the human expertise necessary to assess properly the consequences of alternative data processing policies.

Upon reflection, it should be recognized that this two-fold economic optimization problem posed by data processing systems is merely a special case of the technological and allocative questions facing the general economy. This suggests that an analogous allocative regulated market-type network of competitive consumers and suppliers would have compelling advantages.

In a competitive economy, technological efficiency tends to occur because users have unfettered freedom to gravitate toward least-cost suppliers. Allocative efficiency emerges when suppliers competitively seek the possible profit opportunities signalled by users' monetary "bids" for different [illegible]. [illegible], different subjective evaluations of output effectiveness are weighted by the relative incomes, and hence relative "bidding" capacities of different users. In the [illegible] market context, ethical objection is sometimes raised to this weighting by income since the distribution of income may not be regarded as "just." This objection is not applicable within the university enterprise where the computer funding levels or "incomes" spendable by potential users are determined by top management and, in turn, various levels of administrative "money managers" within the organization. Presumably, these funding levels reflect decisions on relative priorities which should be respected as legitimate exogenous parameters. Hence, if alternative uses of data processing resources are appropriately priced to reflect their costs, the bidding process by users constitutes an effective mechanism to aggregate the weighted effectiveness judgments of users and call forth [illegible] optimal output constellations.

The [illegible] of networking is the promise of the amelioration of some of the conflicts of combined vs. separate and [illegible] vs. centralized. Every computer system is basically a network. We tend to think of a network as involving telecommunications links or at least a significant physical separation between the CPU and the I/O devices. The form that the network takes will probably complement the decentralization philosophy of management. The centralized system will probably feature a single CPU with an array of I/O devices located at the remote sites.

A functional distribution of data processing responsibility leads us to a configuration that essentially replicates the function of the central system several times over. It is not difficult to anticipate its incumbent problems of unnecessarily large telecommunications costs, data redundancies, incompatible access methods, varying levels of terminal support, etc. A geographic breakout of data processing support leads to similar problems although some fail-safe opportunities are available if totally duplicated hardware and control system software are operated at each site.

Viewed against this background, the economic advantages of a distributed, heterogenous network become apparent in comparison both to the divisional scheme and to a highly centralized facility. Indeed, the differences between the essentially autarchic divisional scheme and the centralized system can be understood as principally differences of degree than differences of kind.

The autarchic system, wherein the major divisions are basically self sufficient in computational services, is really a series of non-interactive suppliers offering their users multi-product data processing services under monopolistic conditions. No effective competition exists on the supply side, thus alleviating the pressure for technological efficiency. Further, with respect to allocative efficiency, the potential inter-divisional demands for specialized services are not permitted to interact. As Adam Smith observed two centuries ago, specialization is limited by the extent of the market. This factor is particularly critical in computer services because the technology of particular types of output frequently involves high fixed costs and low marginal costs. Under such circumstances, a service may not appear to be economically viable in the autarchic environment, but emerges as fully justifiable when multi-division user demands can be directed to a single facility.

Mere interconnection of the university divisions, even under a consolidating central management, does permit cross-division demands to be registered more appropriately. Thus, more specialization and an increased variety of computational services would be expected. Also, where scale economies exist, some increase in strict technological efficiency is perhaps possible through elimination of duplicate capacities. How, then, does the network of autonomous supply nodes offer advantages over the centrally directed one?

The answer is that, in a theoretically ideal world of perfect information as to technology and user preferences, there is no advantage to the competitive supply nodes. In a more realistic context, however, critical importance must be attached to the pressures of economic Darwinism and the rewards of effective entrepreneurial behavior. The key factors to be considered here are directly parallel to those relevant in the assessment of comparative efficiency in market type economies and centrally planned socialist economies. If one were truly confident of what to do and how to do it, the market type adjustment process is clearly dispensable. However, in the real world, the apparently crude and at times seemingly wasteful groping of the market is frequently a necessity in order to generate experimentally the information which allows technologies to be compared and the preferences of consumers to be determined. Viewed in this context, the competitive entrepreneurial functions of the autonomous supply nodes loom as compelling advantages. This type of system rewards, and increases the relative importance of, a supply node which either produces an otherwise available service with greater technolog-

[illegible] and responds to an economically [illegible] for some new or modified service. In an [illegible] of considerable uncertainty, with respect both [illegible] developments in data processing technologies and [illegible], some conflict among the entrepreneurs [illegible] of alternative supply modes is both understandable and highly desirable. The function of the market [illegible] identify efficiencies. By analogy with the Darwinian model, unless there are competitive species, one cannot conclude that only the fittest will survive.

In sum, data processing supply epitomizes the circumstances in which [illegible] forms of competitive experimentation [illegible] necessary and efficacious. Competitive entrepreneurial behavior, in direct response to unfettered user preferences, is the best guarantee of technological and allocative efficiency within what is a particularly complex and uncertainty-ridden optimization process. While efficient results are possible in principle under a centrally managed system, a realistic assessment of the informational and behavioral factors involved suggests that the centralized system would not be economically responsive. When the microcomputer is considered in this light, one can make strong arguments for its introduction into many facets of the university's computing complex.

## Network Types

It would seem desirable to develop a definition of networking and investigate how the concept might apply to administrative data processing. We have already noted that every computer system is at root a network. We will restrict our discussion of networking to those situations featuring geographic dispersion of hardware and decentralized software development.

The most common form of network is found on most large university campuses where a central computing system (frequently composed of two or more CPUs) supports an array of remote input output devices. Most typically these devices are CRT, printing or graphic terminals operating at 30 to 300 characters per second and card reader/printer units operating at 1000 to 2000 characters per second. These devices are usually attached via dial-up or leased telephone lines, but where shorter distances (typically less than 1000 feet) permit, they may be wired directly to the CPU, transmission control unit or "front end" communications controller.

Such an environment has the effect of pushing the [illegible] input operations out of the data center and back into the user offices. It is always surprising to see how [illegible] is forgot by many administrative [illegible]. Enough experience has been developed over the past ten years to rather conclusively demonstrate the advantages of such a move — reduced input error rates, more timely production of reports, increased security, etc. The concomitant of such a move is reasonably clear also. Administrative units must begin developing, somewhere on their own staff, some minimal expertise in using and understanding the computer, particularly as it impacts the processing of data under their charge. But more importantly, they quickly perceive the need to develop skill levels sufficient to move beyond bare understanding of the processing of their data to the capacity to develop *ad hoc* reports, alternate output forms and more efficient input techniques featuring something approaching source point data capture. In short, this form of networking encourages, if not requires, the development of some computing expertise within the administrative unit.

While the impact on the user unit may seem striking, the ultimate change in the development of software is more pronounced, albeit more subtle and slower in unfolding. The typical batch system in a non networked environment featured the development of application area specialists in the information systems development group. Interactive, networked environments have changed the focus from the application to the generic processes common to all applications. The information systems development group finds that its resources are more profitably expended on the development of general query systems, data base interfaces, *ad hoc* report writers, and the like. In fact, we frequently see the application area specialist moving to the administrative unit and becoming the nucleus of its computer expertise. As the focus shifts from scheduled, fixed, batch reporting to interactive, *ad hoc* query the need for software support in these areas becomes extremely pronounced. It is not uncommon to see multiple data base management systems and several query/report writer systems implemented on the local network. The level of computing self sufficiency of administrative units on some campuses is the same as in the academic units.

The extramural network was pioneered by the multi-campus university systems. It seems strange that most such networks were built to stifle or inhibit competition rather than to channel it for the common good. In view of our general feelings about a vigorous, competitive economy [illegible] and [illegible] generally expressed abhorrence of monopolies [illegible] [illegible] [illegible] straight [illegible].

During the sixties, governing boards and state boards [illegible] looked only at costs, not at the benefit, [illegible] [illegible] in higher education. Computing is not only [illegible] of financial resources, but should offer the be[illegible] [illegible] for more effective resource allocation in th[illegible] [illegible] of reduced public and legislative support of h[illegible] [illegible] [illegible]. We might venture the hypothesis that it [illegible] effi[illegible] cient allocation of computing to the administrative tasks of the college and university that offers the greatest challenge to our institutions in the [illegible].

Early extramural networks were designed to reduce hardware expenditures in response to "Grosch's Law." They generally featured high speed, remote input/output stations on "branch" campuses communicating with the central system located on the "main" campus. Leaving aside the pros and cons of such an arrangement on academic computing, the impact on administrative data processing has been [illegible] [illegible]. From these early efforts came the interest in developing common software for administrative data processing. The arguments continue to rage over just how "common" registration is on campuses of the same university system or how "common" they should be.

On the positive side came the work of the National Center for Higher Education Management Systems (NCHEMS) in developing basic definitions and category coding for administrative activities in higher education. Also on the positive side, but more as a reaction to mandated sharing, came CAUSE, organized to facilitate the sharing of administrative software and software development experiences.

This early form of extramural network was frequently successful at the cost of failing. Computer hardware costs were held down at the expense of technological growth and innovation. Mandated, monopolistic software systems were installed without adequate system design and have been perpetuated by local patching until there is more that is different than is the same between campuses. Batch systems built around scheduled reporting have continued to be installed long after the advantage of interactive systems became apparent. Centralized development groups have perpetuated the "let us do it for you" attitude of the 1950s leaving many administrative units feeling it was done "to" them rather than "for" them.

[illegible] expanded [illegible] to the [illegible] in hardware and [illegible] feature [illegible] messages, per- [illegible] Typi- [illegible] homogeneous [illegible] communication protocol. [illegible] has of necessity [illegible] decentralization has [illegible] sections of a [illegible] a local interface [illegible] Alternatives to [illegible] systems devel- [illegible] networking [illegible] comprehensive [illegible] software houses specializing in [illegible] as well as a host of software packages [illegible] meet their own particular needs, [illegible] high in the [illegible]

[illegible] on net- [illegible] The Advanced [illegible] network stands out as [illegible] The heterogenous [illegible] communication [illegible] interconnection of [illegible] forced [illegible] to top priority. [illegible] machines the [illegible] error identification and [illegible] From these early [illegible] packet switching, [illegible] handling, and the network inter- [illegible] the local host machine from having to [illegible] network.

[illegible] network [illegible] as its *raison d'etre* [illegible] computing machines. [illegible] networks [illegible] there tend to develop [illegible] hardware architectures may [illegible] network may develop particu- [illegible] software areas. Experience with [illegible] has been generally limited, e.g., [illegible] connects only about 40 institutions of [illegible] slow movement toward general [illegible] this form of networking can probably be [illegible] administrative parochialism — the fear [illegible] This attitude [illegible] on the part of [illegible] keep all the work at home [illegible] that anyone on [illegible] to predict that we will [illegible] networks in the [illegible] the communication of [illegible] independent of the vendor hard- ware [illegible] expect to see this form of net- [illegible]

The [illegible] network actually encourages the [illegible] software by allowing the rout- [illegible] target machine where processing [illegible] expect to see a further breakdown of [illegible] systems concept of software development and its [illegible] "[illegible] kernels" designed to do very spe- [illegible] The [illegible] portability of code [illegible] the center of concern, but rather the trans- [illegible] data to be processed. A later section will [illegible] a national computing network [illegible] well as the [illegible]

[illegible] administrative data processing are [illegible] together [illegible] Certainly the development of [illegible] suppliers for financial, personnel, student [illegible] systems is possible. Whether they will [illegible] whether they will be used, [illegible] The combined vs. separate [illegible] new dimension in the context of hete- rogeneous network [illegible] the argument frequently advanced that we need a [illegible] and "X" machine for our administrative work [illegible] dictate the choice of computer on the local campus [illegible]

## The Impact of the Micro Mini

The advent of the low cost, high capability micro and minicomputer suggest other configurations for the computing [illegible]. Processing capacity and data may both be distri- buted through a network to the location where most effi- [illegible]. In order to facilitate data interchange through the network there might be some central switching node or the network might feature some form of message processor at each node. It does not seem likely that the mini will supplant the big machine, but rather complement it in some network environment. This scheme seems to be the heart of IBM's Systems Network Architecture and is the [illegible] strategy is being pursued by most other vendors of [illegible] systems. The widespread use of teleprocessing [illegible] online or interactive systems marked the fourth generation. Then the spread of the ubiquitous mini and micro [illegible] computing networks will mark the fifth.

The micro computer is available in chip form for under $100 and in a complete desk top unit with tape, floppy disk, printer, CRT and keyboard for from $2000 to $20,000. The micro is characteristically a single user system which substitutes manual user intervention for some of the more complex software functions of the operating system on large machines. The supervisor and language interpreters are usually in some form of read only memory. This permits nearly all of the 16K to 256K bytes of the main memory to be occupied by application programs. The cycle speeds compare favorably with medium sized CPUs of the IBM [illegible] class. Most of the micros use some form of BASIC or APL as their base programming system which places the user in an interactive programming environment. Current research indicates that software development is accelerated by a factor of 10 to 100 over more conventional batch programming languages and techniques. The price of micros can be expected to continue to plummet downward while their capacity and sophistication continue to expand.

The minicomputer is typically marketed as a multi user system priced in the $30,000 to $150,000 range. Typical configurations support from 16 to 32 users in most standard programming languages (particularly COBOL and FORTRAN). Most systems feature some form of supervisor which implements task switching and interrupt processing while holding system overhead to a minimum. Cycle times on many minis compare favorably with machines in the IBM [illegible] 158 class. Nearly all major manufacturers of minicomputers offer some form of data base management system and the capability to attach a large number of high performance "Winchester" type disks. Most systems are designed as sophisticated time sharing machines with a background batch controlled by the supervisor. At the high end of the minicomputer spectrum it is basically only this design feature that separates them from other mainframes.

## Software Sharing

Perhaps the most important impact of networks on administrative data processing has been the realization that software development is the all important dimension. For any significant level of software sharing to take place, it is [illegible] , to [illegible] and [illegible] . A[illegible] in [illegible] has need to [illegible] [illegible]ally. [illegible] this [illegible] that a sequence [illegible] etc. [illegible] have been promulgated and accepted [illegible] institution.

[illegible] it becomes necessary to define, [illegible] a student record, an accounting [illegible] level [illegible] tend to see the dif[illegible] rather than sameness [illegible] of institutions. The [illegible] and [illegible] upon a basic student [illegible] is still before us. [illegible] systems have subscribed [illegible] and have discovered, as a consequence, that administrative systems can be transportable. What [illegible] these institutions is the separation of input, [illegible] management and reporting to provide the [illegible] hardware data base software indepen[illegible] computing diversity to exist [illegible] sharing of software.

[illegible] software sharing requ[illegible] the [illegible] system or some "minor" modifica[illegible] major vendors of administration software [illegible] rather than modules. Just as with hardware, [illegible] the impact of networking to force the devel[illegible] software components as a complement, if [illegible] for the total systems solution. Hopefully, [illegible] the next generation of administrative software in [illegible] Exchange Library begin to reflect this move [illegible]. The cost of developing a reasonably [illegible] student records system for a major uni[illegible] the hundreds of thousands of dollars. The [illegible] life [illegible] such a system is hardly even more than [illegible] years. The cost of trying to limp along with an obso[illegible] system may be even more than re-design. Given these [illegible] parameters it is obviously necessary to separate the hardware development features of such a system from its basic processing definitions which hardly, if ever, change.

One dimension of the technological obsolescence problem [illegible] yet to be faced is the programing language dilemma. The vast majority of administrative software is developed in COBOL, a language developed in the context of first generation computing technology. The language is oriented to batch processing, difficult to learn and not particularly efficient in execution. The prospect of [illegible] a modern, interactive, more powerful language is suf[illegible] to send most programming managers into a state of [illegible] the [illegible] nevertheless, we have to recognize that the [illegible] beast needs to be put to pasture. Perhaps, the [illegible] generation of administrative systems will [illegible] the [illegible] of batch languages on systems development.

## The Minicomputer in Networks

The past several years have brought explosive growth in the use of the minicomputer in administrative systems. In general, the interest in minicomputers has been fostered by the development of the local network. Obvious candidates have been the library, bookstore, power plant and print shop. In these environments there have been significant financial advantages for source point data capture or automatic control systems.

Both the library and bookstore have cost effective applications for wand reading used in check out and inventory cont[illegible] in each case there exists the need to communicate with the central computing facility to interface with the university purchasing system or a master data base. In these energy conscious days the advantages of sensor based control and management of the heating and air conditioning systems has large monetary payoffs. Linking the institutional text editing systems with the printing plant offers obvious economies in re-drafting and re-proofing published text.

In nearly all cases, a communications link with the central facility is necessary in order to update a master file in batch mode, to provide an archive or back up for local files and frequently to facilitate loading the files of the minicomputer. A common operational mode is to "down load" the files or portions of the files from the campus facility to the minicomputer. At night, the minicomputer spools its transactions to the campus facility where the master data base is updated to reflect the day's transactions. This process is repeated on a daily basis. The minicomputer is used in something approaching a real time control environment. Scheduled applications programs are run against the master data base on the campus facility.

In these environments the minicomputer is usually controlled by an operating system, supporting eight to 32 terminals, printers, wand readers, etc. The data base management system on the minicomputer is likely to be different from that used for the master data base on the campus facility. The minicomputer is not used to solve the [illegible] stem problems of the libra[illegible] [illegible] bookstore, but [illegible] [illegible] limited subject [illegible] dealing with control and [illegible] [illegible]. In many of these applications the mini [illegible] [illegible] [illegible] assembler on the central campus [illegible] and [illegible] programs as well [illegible] loaded from the central computer.

The use of the minicomputer in this type of heteroge[illegible] network environment can be expected to increase dra[illegible] [illegible] during the 1980s. It is easy to envision a [illegible]computer in each administrative office handling the [illegible] daily transaction analysis and communicating with the [illegible] [illegible] [illegible] [illegible]. Software development is [illegible] [illegible] in this environment where lan[illegible] [illegible], data base management systems, de[illegible] support, [illegible] are [illegible] varied and subject to frequent [illegible]hange. We would [illegible] to see a move toward the devel[illegible]ment of real [illegible] expertise on the staffs of the administrative departments. Probably, the minicomputer facility will be [illegible] [illegible], independently of the central campus facil-[illegible]

## The Role of the Microcomputer

Perhaps the [illegible]gest impact on administ[illegible] data pro-cessing during the next decade will come from [illegible] prolifer-ation of the ubi[illegible] micro into the local and extramural network. It is likely that the majority of terminals being acquired contain a microcomputer base. The step from firmware to softwar[illegible] [illegible] is simple enough technolog-ically. The questio[illegible] [illegible] [illegible]ndors is when will the [illegible]-sumer switch [illegible] [illegible] [illegible]er "smart" terminal to the programmable [illegible] [illegible] [illegible]udging [illegible]om the phenomonal success of Radio S[illegible] [illegible] Apple, the time is now.

The micro[illegible] [illegible] a level of sophistication on the part of the [illegible] exceeds that currently required of [illegible] terminal operators. Whether the threshold to reach [illegible] level is too high is currently being vigorously debated. [illegible] narrowly defined operations the micro can be "down [illegible]ded" over a commun[illegible]tions channel and supported with a [illegible] "bullet-proo[illegible]" man machine dialogue. In those [illegible] where the narrow definition is possible the mi[illegible] is [illegible] cost effective if only measured in terms of the [illegible] it removes from the large central facility. It has the additional advantage of not putting the whole system out of commission when it fails.

One significant advantage of the micro over a terminal system comes when the user is capable of performing mini-mal programming tasks. Simple, but well designed, file systems, report writers and text editors amplify by orders of magnitude the flexibility and usefulness of the micro. In the hands of a knowledgeable user the capabilities of a [illegible] are almost unlimited. When the user has a programmable work station it is a simple matter to add mul-[illegible] plotters, digitizers, special input pads or a host of other graphic and labor saving devices. We expect to see microprocessor based workstations commonplace for profes-sionals, engineers, statisticians, architects, etc. The supply of computer programmers and hardware designers will probably determine whether the ubiquitous micro enters computing networks as programmable or an exceptionally smart application-oriented terminal device.

The special purpose micro is the heart of word pro-cessing equipment now flooding the market. Many users are discovering, much to their chagrin, that their appetite for word processing quickly encompasses the need to access other computer files, a capacity not available on the stan-dard word processing systems now on the market. It is difficult to make a case for acquiring any micro-based stand-alone equipment which is not user programmable and does not support a communications interface.

# THE FUTURE

"If the only tool you have is a hammer,
you tend to see every problem as a nail."
Abraham Maslow

## Administrative Entropy

About the turn of the century a very clever English physicist, Clerk Maxwell, suggested a way to beat the Second Law of Thermodynamics.[83] The Second Law has to do with entropy — the measure of the unavailability of useful energy in a system and its increase with time when the system is left to fend for itself. Maxwell said, in effect (this is a very liberal paraphrase): Imagine a Demon, a small but very intelligent creature who can see molecules. Now imagine the Demon as the custodian of a gate between two containers of gas at equal temperature and pressure. By careful opening and closing of the gate the Demon allows the faster molecules to move to one container, the slower molecules to the other. Over time, one container gets hotter and the other cooler, thus increasing the available energy in the system as measured by the temperature

---

[83] Much of this chapter appeared in R. C. Heterick, "Administrative Support Services," *CAUSE/EFFECT*, November 1981, pp. 28-32.

differential between the two containers. This increase in energy is accomplished by adding no new energy to the system (other than our smart Demon) and we have circumvented the Second Law.

We do not propose that we can repeal the Second Law of Thermodynamics, but we do believe we can utilize technology as our demon and more effectively tap the latent energy in our institutional system. It seems that universities are ripe for the introduction of a demon of their own into their management philosophy and structure. It is the intent of this section to suggest what form such a demon might take and how it would impact administration in higher education institutions.

These comments are predicated upon two major observations regarding the climate for higher education between now and the turn of the century. First, we can observe a prolonged period of reduced Federal, State and perhaps even public financial support for higher education. This situation will be greatly exacerbated by continuing double [illegible] on colleges and universities will be severe. [illegible] for new construction will be increasingly difficult to secure and physical plant maintenance, particularly for older structures, promises to be exorbitant.

The ability of institutions to react to new educational service opportunities will be significantly attenuated. Such a loss of flexibility will be particularly pronounced and potentially devastating for many institutions. The dynamics of higher education are moving inexorably away from intensive, resident instruction, partially due to the same economic considerations impacting our public as well as ourselves.

As we increasingly view entire states and even larger geographic areas as our constituency and perceive the need to bring our educational expertise to the people, the cost of doing so seems to be increasing exponentially. More and more the university will be, rightfully, viewed as a network geographically dispersed and increasingly reliant upon effective administrative and public communications. This brings us to the second major observation. Institutions are becoming noticeably more information intensive. Our role has always been to discover, develop and disseminate information but our internal operations have never been so dependent upon effective information purveying as they are now, and increasingly will be. As the institution grows in complexity, as it continues to reach out geographically, as its involvements in sponsored research and public service reach higher levels of activity, the demand for prompt, [illegible] information, and its cost effective dissemination [illegible] the [illegible] administrative structure, grows [illegible].

[illegible] we have [illegible] attempts to address these [illegible] related problems while reducing administrative [illegible] a percentage of total costs, through satisfaction [illegible] this increasingly [illegible] appetite for [illegible] management information and information-related academic and administrative support services. The [illegible] of this [illegible] is devoted to suggesting some ways and means that information technology and management science [illegible] be applied to [illegible] our administrative processes and services while [illegible] line, or even reducing, the percentage of the budget [illegible] to administrative support services. The suggestions [illegible] by no means exhaustive, [illegible] not even the [illegible] pertinent ones that might be [illegible] but they will [illegible] a flavor of what is possible and [illegible] portray faithfully the exciting opportunity [illegible] colleges and universities in this next decade.

[illegible] which [illegible] this future with candor [illegible] to be adaptive if [illegible] will have an exciting opportunity to propel themselves considerably higher in peer and public perception of leading quality institutions. Such a move will require that the institution be prepared to react quickly to, or even to [illegible], new opportunities and changing educational environments.

Many institutions are experiencing significant improvement in their faculty resource. We are not at all convinced that our administrative support services have kept pace with the improved quality and entrepreneurship of our faculties. Our perception of many leading institutions is that they suffer from this problem. The time is propitious for a bold innovative move to address this apparently common malady. Those institutions that move quickly and successfully to bring their administrative support services to the level of their faculty and staff potential should be most favorably positioned to use these coming lean years as a springboard to greater recognition and service.

This section is also predicated upon the presumption that many universities have experienced an increase in entropy — an increase in the under-utilization of their useful energy. There is an ebb and flow in all organizations and, of course, no organization ever captures 100% of its latent energy. Our demon [illegible] be employed to release this latent energy, channel it toward our institutional pur-

[illegible] and create an adaptive organization utilizing sufficient [illegible] to continually regenerate the [illegible]. In computer parlance we need a system that does things for people rather than a system that does things to people. The opportunity exists to turn our focus from administrating to providing administrative support services. To such an extent the [illegible] administration can concentrate on doing the right thing while the [illegible] organizational energy [illegible] on doing the thing right.

We view the university as a system — a system composed of *purpose*, *parts* and *structure*. The *purpose* or [illegible] of the institution is reasonably clear and continually honed to sharper focus by its top administration. The *parts* are its people resources, its physical plant and [illegible] potential. The *structure* of the system must [illegible] a symbiosis between the parts and purpose; the structure must provide for coupling among the parts and between the parts and the purpose. The glue or [illegible] of this coupling is decision-influencing information. The role of administrative support services should be to provide a commodious (certainly non-hostile and non-threatening) physical environment in which pertinent, prompt, reliable information is exchanged [illegible] exchanged because both positive and negative feedback mechanisms are important to both the parts and the purpose.

The key to successful administrative support is an open, inviting, facilitating physical environment in which necessary information for decision making, at all levels, is freely exchanged and readily available. What follows are some random observations on how such an information intensive administrative structure could be developed.

## Administrative Structure

Clearly, the organizational structure sends signals to its constituent parts and either facilitates or filters information [illegible]. The signals emanating from many structures [illegible] administrative offices are mixed. On the one hand, many of these offices have regulatory or at least approval responsibilities [illegible] responsibilities tend to focus concern external [illegible] organization. We have in mind offices such as Personnel, Purchasing, Accounting, Health and Safety, etc. where dominant concerns are State and Federal regulations, internal and external auditors, etc. On the other hand, these offices control the dissemination of information that is the very life blood of [illegible] reports [illegible] principal [illegible] information is frequently [illegible] often of little decision value [illegible] difficulty in acquir[illegible]

[illegible] open information dissemina[illegible] frequently obscure and unin[illegible] regulation is enough to make anyone [illegible] A strong case might be made for [illegible] approval function from the infor[illegible] function. Whether such a structural [illegible] desirable seems to be more a [illegible] style and philosophy than [illegible] In any event, the need for att[illegible] to [illegible] most reliable data for use in decision making [illegible] of this section.

[illegible] standpoint of information dissemination one [illegible] signals. Some of the information dis[illegible] offices on most campuses would [illegible] the learning [illegible] centers, print and [illegible] center. Each of [illegible] areas of experience and [illegible] goal of cost effective infor[illegible] dissemination. On many campuses some report [illegible] channels, some through administrative [illegible] A strong case could be made for structuring the [illegible] these offices reported through the [illegible] the synergy necessary to seize [illegible] an information intensive university. [illegible] be predominantly a question of man[illegible]

[illegible] structure is settled upon, it [illegible] information availability and inter[illegible] an administrative concern and goal. The uni[illegible] at the cutting edge of technology and [illegible] draws people into its parts who are excep[illegible] [illegible]able and highly motivated. The problem is not so much to manage these people (in fact, this is [illegible] the wrong thing to do) as it is to manage [illegible] and communication links that tie [illegible] Administrative support services should be [illegible] a barrier free environment for information [illegible]

[illegible] administrative structure of the institution should [illegible] the renewal of latent energy in the system in [illegible] providing the [illegible] coupling for the [illegible] needed [illegible] organism the [illegible] to admit and [illegible] periods of [illegible] the next [illegible] promises to be just [illegible] higher education the necessity of an adaptive [illegible] organization [illegible] efficiency.

## Administrative Innovation

[illegible] the physical environment [illegible] physical plant [illegible] be separated from the communication [illegible] because many of the communication channels are themselves physical, it seems appropriate to consider the physical [illegible] as part and parcel of an information intensive administrative support system. During this next decade [illegible] of a [illegible] and facilitating physical environment [illegible] to be one of the most difficult portions of the [illegible] network to keep in place. Most campuses are [illegible] its current use not particularly adaptable [illegible] it will be pressed to serve. Much thought and [illegible] need be placed on renovation and adaptability [illegible] space as funding for new construction will be exceptionally difficult to acquire. As energy costs assume a la[illegible] share of the maintenance budget the desirability of a [illegible] energy management plan and energy manag[illegible] network grows. To the extent an information intensive administrative support system can be put in pla[illegible], the need for expanding administrative space requirements should be obviated. Controlling the growth of space required by administrative functions can help relieve what promises to be one of the most critical problems of the next decade for many institutions.

The university network obviously includes physical access; parking, vehicular and pedestrian travel facilities; physical containment; new construction; maintenance; and a plethora of other concerns that will be taken for granted here so that concern may be focused on several new requirements.

Underlying the capability to effect many of the ideas that will be developed is the presumption of the existence of a local communication network. The campus already has several different networks in operation — the most taken-for granted is the telephone. In addition, most have the start of a cable television network and an exponentially expanding computer network. It is easy to envision a [illegible] network [illegible] The local [illegible] [illegible] of [illegible] mentioned [illegible] like those [illegible] cable [illegible] terminal [illegible] computer [illegible] at the [illegible]

[illegible] network for most campuses [illegible] hundred [illegible] thousands of dollars. [illegible] information technology [illegible] network of this type.

[illegible] within our capability to replace nearly all [illegible] accounting, purchasing, personnel, etc.) [illegible] entered directly from the departmental [illegible] and approval can be done [illegible] back [illegible] of paper [illegible] Even more use [illegible] of everyone involved keeping [illegible] of the same transaction. The need for [illegible] and Principal Investigators to keep [illegible] might be eliminated. A [illegible] electronic mail and message [illegible] filing system. Document storage on [illegible] from [illegible] to 100,000 times less than [illegible] than current paper filing systems. The [illegible] the library are obvious [illegible] the potential [illegible] literally hundreds of campus admin[illegible] more subtle, but equally significant. If [illegible] cabinet requires about four square feet of [illegible] to find about 30 of them in each [illegible] departments, they require something on the [illegible] square feet of floor space. If we expect the [illegible] to maintain something like three times [illegible] of the [illegible] departments, then we are consid[illegible] nearly [illegible] square feet of floor space occupied by [illegible] cabinets. This could be traded for a mass storage [illegible] roughly 100 square feet. The net saving [illegible] fair sized building.

[illegible] typists and clerical personnel is [illegible] use of an effective text and document pro[illegible] and retrieval system, and we consider that our [illegible] administrative offices utilize the services of [illegible] such people and each such person occupies approximately 100 square feet of floor space, then another 25,000 square feet of space is available for expansion. We will omit additional examples, but the potential for improving our internal communications, increasing our effectiveness and controlling administrative costs is truly significant.

We would have to confess that the foregoing examples, and literally hundreds of others we might mention, are rather pedestrian. Let us consider several examples that have the potential of changing significantly our operational habits. Proper application of technology, as we said earlier, will not only change how we do things but also what we do. It is a simple (though not inexpensive) matter to provide library searches from a faculty member's desk or a student's dormitory room. Combined with an effective delivery system the vehicular and pedestrian traffic patterns and space utilization aspects of the campus could be significantly changed. Even more radical changes are possible if we contemplate the prospect of placing the contents of books and serials in machine readable format. On many campuses the faculty have already begun making their lecture notes available to the student via computer text files.

Every parking space (say around the library or the computing center) requires approximately 150 square feet. When users can access the services of the library, or what have you, without the necessity to physically travel to and park their vehicle, tens to hundreds of thousands of prime, central square feet of space on campus can easily be freed not to mention recapturing the personnel time lost in transit.

University publications could be taken directly from text entry to the final print shop production electronically. As the percentage of copying of computer output grows (by the end of this decade it will represent 80% of total copying) it will be necessary to connect the copy centers to the computing center. A large volume of the graphics currently being produced on campus comes from the computer. It is possible to produce slides directly from the computer. In fact the bulk of commercial illustration (nearly all TV commercials for instance) are produced by computer. During the next ten years most illustrative work on campus will switch to being computer generated. The integration of the computer, television and the learning resources center is an untapped potential on most campuses.

The theme of this section is information technology and administrative support services. We feel, quite strongly, that the future of higher education is intimately intertwined with information and its effective communication — a statement that appears so obvious as to be trite. But it is written from the viewpoint of asking which future developments might significantly impact our capability to grow in both service and quality. An exceptional opportunity exists to better utilize the capabilities and energy of our faculty and staff by making available information to improve their decision processes. This can be accomplished by putting in place a truly effective administrative support system based on careful analysis and anticipation of future developments in information technology. Such an environment will come. The question is will we shape it or simply be carried into it — can we seize the opportunity as a spring board to a higher level of service to our current and future constituencies?

## Organization

The distribution of computer competence and sophistication in any institution of higher education is extremely skewed. A look at our short term (two to three years) future may be had by simply observing the current activities of the most advanced ten percent of users on our own campuses. If this skew does not exist among campus users it is a pretty fair sign that computer operations are stagnant. If the skew does exist we should not be surprised to see the top five or ten percent of users leading even the computing operation. It is the pressure applied by this group that propels (or drags, depending upon your point of view) institutional computing activities into the future.

Of some concern is the question "Can we control, or at least shape, our computing future?" It is exactly this concern that leads many computing directors to focus what is considered by many an inordinate concern on this ten percent of the user community. While the dynamic nature of this institutional computing enterprise may make for an exciting environment, it also makes for a disquieting one for those of us who require stability in our computing environment. We need to be careful not to counterpose "dynamic" and "stability." Stability can be achieved in a dynamic environment if the constant procession of change can be made essentially transparent to the user. To achieve stability at the expense of growing with the technology is much too high a price to pay.

One method for coping with such a dynamic environment is to develop modular software which separates the interface to dynamic hardware and technology from the stable processing logic. If this is to be accomplished it means that the next generation of administrative software systems will need to be conceptually re-designed as well as re-programmed. The burden of this effort will need to be shared much more equally between the Information Systems group and the User Departments than has been the case in the past. Controlling this dynamic environment will require re-thinking the role of the central campus facility as well. There will need to be significant charges in management structure, operations procedures, network design and intra-institutional communication processes. We intend to make this arbitrary separation of topics for further discussion. It should be borne in mind however, that they are all part and parcel of a comprehensive, integrated development for the future.

We believe the next ten years will see a major restructuring of the organizational chart of many institutions of higher education. As institutions begin to recognize their increasing dependency on computer related activities for their day to day operation, and as the identifiable computing related share of the budget exceeds ten percent, the pressure for a more consistent, global view of computing will become insistent. We see two possible scenarios as the most probable organizational response.

The first scenario, and we think least likely, presumes the widespread acceptance of microcomputer by users and user willingness to develop a reasonably strong capacity to use and program them. It does not seem unreasonable to expect 1000K and up main storages and cycle times of 100 nanoseconds in the next two generations of micros. Should the user community decide to move in this direction it is logical to predict the reduced impact of the central facility on user processes and the consequent attenuation of the importance of computing centers in the general scheme of educational institution management structures. In such an environment we can expect to see the Information Systems group assume an operational role much on the lines of most other administrative departments the campus print shop or power plant might be good comparisons. In this event we would expect to see the computing operation under an Administrative Vice President at a reporting level such as Buildings and Grounds, Central Stores, the Print Shop, etc.

Far more likely, at least for the next decade or two, will be the desire of top management to centralize, focus on, and attempt to capitalize on, information related activities across the entire spectrum of the educational institution. In such an environment we would expect to see the amalgamation and consolidation of information related activities resulting in a re-alignment of vice presidencies and the creation of a Vice Presidency for Information Systems. The re-alignment we think most likely will involve:

Computing Center
Library
Learning Resources Center
Print Shop
Copy Center
Mail Room
Telecommunications Center
Graphics and Micrographics
Centralized Word Processing
Archival Records

Such a group likely accounts for 10-20% of the institutional maintenance and operations budget. With the exponential growth of the use of information technologies this figure (depending, of course, on accounting procedures) can be expected to double in the next 20-year period.

With either scenario, we expect to see greater user autonomy and independence. The user will insist upon his computing being done in an environment paralleling the way he uses the telephone network. He will also insist upon the same level of transparency as to how his processing is actually effected. The demise of mass transit systems in favor of the personal automobile was heralded by the advent of the highway network. Computer communication networks are likely to have the same impact on user preferences as to the style of and environment in which he does his processing.

The impact of either scenario on the computing operation is not likely to be so pronounced. The need to operate a central computing facility will continue, although we expect to see many, if not all, of the labor intensive input output operations move out of the central facility and into user offices. Correspondingly, we expect to see a much expanded role for the user services division of the computing operation. At the same time will come a much stronger focus on maintenance and other field service operations. Much of this new emphasis will be directed toward the teleprocessing and network aspects of the computer operation. The software development arm of the information systems group will become more involved in developing broadly used software products as opposed to the current, more specific, focus on developing a "total" system for a given user office. A much more equitable balance of hardware and software talent can be expected to emerge in all computer facility groups.

## Computer Operations

The future should bring major changes in computer operations. We expect to see the central computing facility move rapidly toward becoming the "BIG SWITCH" in the campus information systems network. More and more the computing center will serve a facilitating role, less and less a computing resource role. The predominant load on the central system can be expected to be network control, data storage and switching.

The central facility will become only one node in network of information processing facilities. It will feature those very large, highly shared resources which would not be cost effective if not shared among a large number of users. The central facility will contain large mass stores (over one trillion bits) which will be used as medium term (2 weeks to 2 years) archives and as backup for a whole host of software and data distributed through the network. In the near term, it will provide large scale (20,000 lpm) laser printing, although we would expect this activity to be assumed by the campus print or copy shops over the longer run. In the shorter run, we would also expect to see the central facility develop some in-house micrographic capability if it does not already have it.

The computer resource that will need to be supplied by the central facility will be high speed "number crunching." Campuses without a heavy research orientation may find it cost effective to acquire this service, on those occasions when it is needed, from some external supplier. Even when the campus facility supplies this service there will be need for expanded linkages from the central system to outside computing sources. It should be possible for any on-campus user to connect to the central system, and through it to effect a high speed connection (50 KB or higher) to some other computer system not located on the campus.

As the central facility moves into this "BIG SWITCH" mode it will be necessary to develop much higher volume communication channels than are now the norm. This is particularly evident if the central facility is to serve as a repository for large data bases and as a short to medium term archive for decentralized data bases and software systems. With presently foreseeable technology and costs this will require wiring the campus with coaxial cables. Such an effort will be expensive but have major payoffs to the information systems network. It will permit transmission of graphics, television images and electronic documents as well as computer data.

The operation of the central facility will become more and more like the operation of a large power utility. The concern will not be with specific users or specific tasks but rather with total load and total demand. The Computing Center will not only run user jobs but, in fact, will not even be aware of what is running on the system. Just as the power company sees a home not as a collection of appliances and their uses but rather as a consumer of amperes, so also the computer center of cycles, memory and storage. Computer operations will change from management of computer work to management of flow.

## Administrative Systems

[illegible] we have said that universities need to see their [illegible] in a [illegible] of [illegible]; that universities are, and will [illegible] information intensive organizations; that changing [illegible] can best be [illegible] by a modified organi[illegible] structure that allows the organization to achieve [illegible] and [illegible]; that the physical [illegible] parts of our communica[illegible] and [illegible] them can be directed [illegible] and information delivery [illegible] must be structured to [illegible] under a single [illegible] that [illegible] attention to [illegible] be in a better position to meet the [illegible] and challenges of both today and tomorrow.

With that as a backdrop this section will elaborate on how the administrative processes themselves are likely to change. We assume that there will be greater "connectivity" through the implementation of local networking and that there will be quicker communications due to the connectivity. The first impact, we believe, will come in the substitution of paper flow by electronic document flow. Systems will have to be designed that not only capture the contents of the document presently in use but also capture its flow through the organization. Just as most offices time-stamp all incoming mail, so also most electronic transactions will be time-stamped in order to provide a better tracking mechanism. Further, independent of the data contents, there will of necessity be a complete and independent audit trail so that actions by the system can be verified, audited and reconstructed if necessary. It will be necessary for software systems to be able to reconstruct a "data picture" and a transaction flow at any point in the past. Complementing the ability to go to any point in the past, it will be necessary for software systems to be predictive so as to be able to generate a "data picture" for any time in the future.

At the present time, the major obstacle to a complete replacement of paper documents by electronic means is the necessity presently placed on signatures. The signature was designed as a means for substantiating written documents. Its use in the electronic age is very limited. The electronic signature verification machine, we believe, is not the correct approach to validating electronic documents. This simply takes a method of verification from an older technology (paper) and tries to force it to work in a newer technology. Fortunately, the signature barrier is breaking down. Stock market orders are executed by telephone calls. Salary checks have computer printed signatures and electronic fund transfers have no signatures at all a secret password is all that is used. Institutions will have to accept alternate means of satisfying audit requirements. Signatures on documents can no longer survive as the accepted means for verification as the originating documents themselves are being eliminated.

It is obvious that means of input into the computer system are changing. Cards gave way to optical forms which in turn are being replaced by terminals. The output side is now ready for some dramatic change. As memory devices get smaller and smaller and as more and more individuals acquire personal computers it is very likely that each one of us will carry with us a memory bank, perhaps the size of a credit card which will plug into the various computer outlets for the computing of data specific to us. This memory bank will then be taken to our home devices for outputs analysis or review. Take a specific example. The university distributes various documents such as class tickets, grade reports, etc. by printing and mailing the documents. We will call this an "active distribution" mode. As more and more individuals have their own devices and can carry around small memory banks that plug into computer outlets, the mode of distribution will become "passive distribution." On a certain date the university will make available the grades or class tickets. Students will either use their credit card size memory banks with the computer outlets to capture their data or simply dial into the computer system and get it online. A clear shift will take place in the responsibility for distribution. At present the University sends out the documents, in the future the information will be made available and the users (students, faculty) will have to access it if and when they need it.

A widespread use of this concept of information distribution will require a change in our treatment of security. Security systems will have to be modified dramatically. At present due to the fact that only a limited number of individuals have direct access to computerized data, security measures can be placed at a task level. That is, individuals, for example, may be permitted to work with personnel data but not, say, with student data. However, as everybody begins to have access to the data bases it will be necessary to provide security not only on the tasks that an individual is permitted to perform but also on the subset of data that the individual is permitted to review and modify. Thus an individual may be permitted to access personnel data and not student data implying task security. Further, the same individual may be limited to looking at personnel data that belongs only to his department. Thus the security is further subdivided within the task and is determined not by the function being performed by rather by the data being accessed. This level of security, of course, is much harder to implement than the previous level. Newer systems will have to be developed to handle the new requirements for security.

The concept of passive distribution of information can be extended a little farther to suggest that we are heading towards a cottage industry of data processing professions. Examine the library. In a research library and in most institutional libraries over 40% of the collection is never used. Another 20% is used only once. This clearly leads us to desire a system in which a text is published only when it is first required. The term for this concept is "on-demand printing." At least for modern publications we are not too far away from on demand publishing. Books and magazine articles are written on word processing equipment and hence are in machine readable form. There is nothing that prevents us from simply keeping it in that form till we realize the first request for that material. At that time the user can request the material either in machine readable form or in printed form. If it is requested in printed form the library simply prints a copy and gives it to the user. One may even debate the merits of requiring the user to bring the material. If the user is not required to bring the material back, then the distinction between the library and the bookstore begins to get very blurred. In any case such a distribution scheme places the present role of the publisher in jeopardy. Perhaps the publishers will become more like record companies. We will just have to wait and see.

In any case, it will be possible one way or another to have document delivery systems that are based primarily on electronic media. If documents can be delivered electronically then why, for instance, do typists (not secretaries) and data input operators have to come to work at all. From the convenience of their home they can do the desired work. The home also becomes the place of work, making further use of the physical plant that is already there. The same reasoning can be extended for systems programmers and application programmers and others who contribute services that ultimately result in changing the contents of computer data banks. This cottage industry resulting from working in the home has an added advantage - commuting will be replaced by telecommuting, saving time and energy.

The concept of distributing work and fragmenting the work environment by not only permitting but requiring certain individuals to work from their homes fits well with the idea of a geographically dispersed university. Once we begin to distribute the daily work through the use of teleprocessing, an important communications barrier is broken. Prior to the invention of the telephone, instant (meaning without delay) communication was possible only when people got together. The telephone very successfully broke the spatial barrier. However, as evidenced by ringing phones and busy signals, the temporal barrier still exists. Telecommuting -- or the use of computer and communications -- permits us to eliminate not only the spatial but also the temporal requirement for instant communications. This opens up a new degree of freedom and its impact will soon be felt.

Clearly many things are possible. How desirable these possibilities are to health and well being of our institutions is a matter that our administrators will be called upon to address -- and soon. The course they chart will, in the long run, determine the course of society itself. Higher education must continue to maintain the role of thinkers, experimenters and innovators.