digital

DISTRIBUTED SYSTEMS HANDBOOK

digital DISTRIBUTED SYSTEMS HANDBOOK 1978-79

DISTRIBUTED SYSTEMS HANDBOOK

digital

This book was written and edited on DIGITAL Word Processing Systems (310W, WT/78, and WPS 102). The finished text (on WPS floppy disks) was input to the DECset-8000 computerized typesetting system and, via a translator program, was typeset automatically without manual mark-up.

Printed in U.S.A.

First Edition, June 1978

digital

He took the wheel in a lashing roaring
hurricane
And by what compass did he steer the
course of the ship?
"My policy is to have no policy," he said
in the early months,
And three years later, "I have been
controlled by events."

THE PEOPLE, YES
by Carl Sandburg

FOREWORD

Computers will eventually have as much impact on our lives as the automobile has had. The computer that began as an expensive and difficult-to-use tool for very special problems is becoming widely used in the office and factory, and even the home.

Digital Equipment Corporation is proud to have been part, along with our customers, of many of the most innovative and foresighted uses of computers. From the beginning we have believed that computers should be tools that could be used by people who need information to do their jobs. We have promoted the design of interactive computer systems that can be placed where they are needed. We see the trend toward the increased use of interactive, distributed computer systems as confirmation of our basic philosophy.

We have always believed that what is good for our customers is good for the computer industry, and what is good for the industry is good for us. This has led us to produce a series of handbooks, sharing our insight to various computer topics. We are pleased to add this handbook to the collection and to share our insight into the design of interactive, distributed computer systems.

Kenneth H. Olsen
President

TABLE OF CONTENTS

PREFACE

Today it costs more to fill your car with gasoline than it does to buy a general purpose (micro) computer (a single integrated circuit device). This apparent phenomenon results from 25 years of steady improvement in computer system cost/performance (the cost to perform a given processing function).

As computers become less expensive, they are used more widely. As yet, no bound to ultimate computer cost reduction has been found. If a computer application is *close* to cost-effective today, then it *will be* cost-effective soon.

WHY WE WROTE THIS BOOK

Over the years, Digital Equipment Corporation has created a series of handbooks, each of which shared our insight and experience in a particular computer-related topic.

This handbook is about distributed systems, particularly those that utilize computers. The primary reasons for writing the book were to define a distributed system and to discuss the future of distributed processing.

Distributed systems are nothing new – many of the minicomputer systems we have built serve distributed processing applications. Our avant-garde customers were distributing their data processing long before the first advertisement heralded its arrival.

But the impact of distributed systems has grown to the point where it affects most computer professionals as well as many people who haven't yet had much to do with computers.

In this handbook we discuss many of the subjects that relate to distributed systems focussing particularly on these points:

- the benefits of using a computer system
- who should participate in the system design
- some of the major design problems

Our key concerns are

- how to design computer systems that non-experts can use to do their job effectively
- how data communications systems and computer network systems can link many widely separated individuals

This is the first edition of the *Distributed Systems Handbook*. We plan to publish future editions that incorporate new technology and understanding and that can benefit from response to earlier versions. A reader response card has been included in the handbook, and we really look forward to hearing from readers.

In the past, we made computers for engineers and programmers, and our handbooks were written for, and by, engineers and programmers. But many of our new and important customers are neither engineers nor programmers, because the computer is now a valuable tool for many more purposes than in the past.

So this book is not about the details of programming or computer design. Instead, it's about how computer systems can be used to manage the information that most of us need to do our jobs.

DO YOU NEED TO KNOW ABOUT DISTRIBUTED SYSTEMS?

Do You Use Information In Your Job?

If you are reading this book, the chances are you use information as an important part of your job. You need to receive information from other people in order to plan and execute your own work, and you need to inform others of your progress and problems.

Do You Use Formal Information Systems?

Unless you work in a small enterprise, some of the information you need and produce probably travels through formal information systems. For example, you read and produce status reports. You send memos. You attend regular meetings.

Does the Information Matter?

Does the accuracy or timeliness of the information you use make a difference in the quality of job you can do? Unless information is really secondary to your work, the quality of the information you use is important to you, and the quality of the information you generate is important to others.

If you use information from formal systems (something more than chatting with the person at the next desk), and the information is valuable to your work, you will probably be using a computerized information system in the not-so-distant future, if you don't use one already.

Are Any of Your Competitors Using Computers?

If you're unsure whether you'll ever need a computer, but some of your competitors are starting to use them, then this book may help explain what you're missing. Not all uses of computers are productive. You shouldn't add computers just to "keep up with the Joneses". But you can't expect that the effective use of computers will be obvious to you, nor that a friendly computer salesman will come around and tell you when you need a computer system.

No one knows more about your job than you do. If you learn a little about computers, you may be able understand your own computer needs much better than anyone else could. The fact that a competitor is using computers may be a sign that you should learn more about computers yourself.

Are You a Computer Professional?

An increasing percentage of all computer systems are used for the kind of information automation described in this handbook. If you are a computer specialist of one form or another (systems analyst, programmer, etc.) then you may want to learn more about distributed computer systems.

THE INTENDED READER

We couldn't write a book that explained distributed systems for someone who knew nothing about either computers or the management of medium- to large-sized organizations. So parts of this book are written for professional managers who know relatively little about computers, and other parts are written for computer specialists who know relatively little about management or about organizations.

Few people will want, or even need, to read all the book. For example, the chapter on design techniques goes well beyond what a manager needs to know about distributed systems. Similarly, most computer professionals already know the material in the introductory chapters.

But we hope everyone who will specify, design, or implement distributed computer systems will find some helpful information here. One of our key points is that distributed systems should be developed by a team effort, where the team includes:

1. **General managers.** Computer systems will play an increasingly important role in the management and operation of many enterprises. The design of systems with such widespread impact cannot be left to computer system analysts or any other set of specialists. The managers must participate actively in setting objectives and design goals for the system.

2. **Information specialists.** In addition to managers, there are others in an enterprise to whom an information system is a critical resource and tool. They, too, should be actively involved in system design and specification, because they know best what they do and how they do it.

3. **Data processing and information managers.** Most enterprises now have, or will in the future have, senior-level managers in charge of information systems, just as they have senior financial managers today. They should be part of the team.

4. **System designers and implementers.** Finally, there are technical specialists who do the detailed technical design and implementation of information systems. They need the assistance of the other team members.

Effective information systems will be built when all of these people work together and understand each others' objectives and constraints.

HOW TO USE THE BOOK

This book is neither a traditional-form handbook from which you can read an arbitrary section at a sitting, nor is it a novel that has to be read carefully from the front to the back.

The first five chapters give an overview of distributed systems and introduce the key themes. If you are familiar with computers and management, you may read these quickly. The first five chapters should at least be skimmed, after which the rest of the book can be read piecemeal.

Chapter 6 is an overview of the issues facing a general manager who is considering acquiring a distributed system. Most of the discussion applies as well to the manager of a department that will be using the system. EDP managers, information specialists, and computer professionals also will want to skim this chapter to get some feel for the problems of the general manager and his role in developing these systems.

Chapter 7 discusses the key problems facing EDP managers. General managers will want to read this chapter as well, since it describes some of the most important computer decisions and their impact on the resulting system. Information specialists and programmers may read this chapter to gain more understanding of EDP management issues.

Chapter 8 summarizes the most important system design issues; those which directly impact the productivity of system users. EDP managers, information specialists, and system designers should certainly be familiar with this material. The chapter is not highly technical, and most of the topics are of interest to all general managers.

Chapter 9 gives a general design model for distributed applications. The material in this chapter is very useful in the initial conceptualization and design of a distributed system and should be understood by EDP managers and system designers.

Chapters 12, 13, and 14 discuss some of the technical aspects of distributed systems and are aimed primarily at the system designer.

Chapter 14 is a brief speculation on where the future may take us and is optional reading for all.

Chapter 15 discusses some specific uses for distributed systems. Most of the book uses general information systems, such as a business might use, as examples. This chapter looks at other kinds of systems. The material is nontechnical, and intended for all readers.

Chapter 1
INTRODUCTION

If you exist in a competitive world where data or information is valuable to you and your ability to compete, then you need to understand computers and their impact on your career or enterprise. That doesn't mean you must learn to program or to design electronics. It does mean you should understand the principles and applications of computers so that you can help plan the system most useful to you.

INFORMATION SYSTEMS

In your day-to-day life you are barraged with an overwhelming amount of information. An information system is a planned or methodical way of gathering or disseminating information.

Formal and Informal

Most people get a large amount of information informally. Rather than reading a magazine or getting a memo you will hear about something because of a chance encounter in the hallway, or because a colleague in the next office happens to mention it. Similarly, many work-related problems are solved informally, by talking to someone in the department, rather than going to a formal report or file.

But the likelihood of an informal communication is reduced rapidly with the increase in distance between work sites. Studies have shown that the chances of an informal communication go down by a factor of two when someone is as little as a few hundred feet away, even considering the use of telephones.

Therefore, any enterprise that is big enough to span several small buildings, or is geographically dispersed for other reasons, has to build formal information systems, because the distribution of information can't be left to happenstance. These systems include regular reports, status meetings, formal memo distribution lists, the use of standard internal forms, like Purchase Orders, etc.

Briefly, a computer is added to the system because it lowers the cost, or because it raises the value of the information, or both.

The Price

Formal information systems are neither free, nor do they work perfectly. They suffer in one or more of the following aspects:

- **Cost.** It can cost a lot of money to run a formal information system.
- **Accuracy.** Important data items may be misentered, miscopied, become illegible, be misread, or simply be misunderstood.
- **Timeliness.** Almost all data that is worth having, is more valuable if it is quickly accessible.
- **Proximity.** Formal information systems can turn into information bureaucracies in which you have to work through a sequence of different individuals before you can get to the information you need.

So far, computer systems haven't been very successful at doing anything to help informal information systems. It's usually too much trouble to create the programs, not to mention to find out what really happens.

But computer systems are proving their value in formal information systems:

- **Cost.** As computer system technology continues to decrease in cost, it becomes less expensive to automate an information system than to run it manually.

- **Accuracy.** Many of the error-inducing steps, such as copying data, can be eliminated.

- **Timeliness.** Interactive computer systems can capture data when it is created (enter sales data when the sale is made, for example) and produce reports, on demand, when they are needed.

- **Proximity.** A well-designed interactive system can be used directly by individuals in the enterprise, with little or no help from computer specialists.

THE USES OF INFORMATION

We've discussed briefly the cost aspects of an information system. If information were not useful we could do without the systems. Let's look at some of the uses of the information itself:

- as a product
- to manage an enterprise
- to operate an enterprise
- for competitive value

A Product

For some enterprises, information *is* an integral part of the product. Travel agents, airline reservations, banks, insurance companies, credit card companies – the success of all these enterprises depends on the quality of their information systems. Not surprisingly, much of the initial development in information system concepts and technology comes from work done in these enterprises.

But these enterprises are also somewhat dangerous examples, because the value of the system is so large. Enterprises where the information system is an integral part of the product *have* to make the systems work. Therefore they hire top computer people, develop state-of-the-art system design, and then have top-grade managers to make sure the information systems keep working. What works for an insurance company for whom the computer system is key to success may not work for a manufacturing company, for which the quality of sand-castings is the most critical thing.

To Manage an Enterprise

Even where information is not the product, it is bound to be a major part of the management in any but the smallest enterprises. Much of management centers on developing plans, implementing plans, measuring the impact of plans, and then reiterating the process. All of these activities require and produce information.

Plans are created on the basis of formal or informal information. The plan is based on an analysis of existing data, and judgment about whether that data represents the right way to run the enterprise. Plans are often expressed as changes in that data – sales to rise by at least 15%; all products with a return of less than 20% to be dropped; etc.

Information to Operate an Enterprise

Information is an essential part of running most enterprises (executing management plans). A sales order is received, raw materials are purchased, machines are scheduled, workers hired, production rates set, inventory monitored, costs evaluated, financing sought. All of these are information-based activities.

It may seem that the job is to run a milling machine, or to weld parts together, but how do you determine how many parts to make when the milling machine is set up for a given job, how many parts do you need in inventory to keep the welder busy? Any successful enterprise has information systems that are key to day-to-day operations. In many cases the information system is informal (the foreman knows what's needed, for example) but the system is there nonetheless.

Finally, information is an important part of many individual jobs within the enterprise. The machinist works to a plan and a schedule. There are few skilled jobs that don't require some information.

Competitive Value

A good information system can help an enterprise function efficiently. Having a better information system than the competition can make a direct difference in success in a competitive marketplace; conversely, having a poor information system may lead to an uphill battle even with good products and a good production process.

Some of the major areas in which a good information system can benefit an enterprise are listed below. The examples come from manufacturing, but information has similar benefits to almost all enterprises.

- **Process control.** Product cost is increased if parts of the production process are poorly utilized, or if production quality slips. Information permits key processes to be monitored and controlled.

- **Clerical productivity.** Clerical costs such as order processing are part of the cost of the product. Lower clerical costs (higher clerical productivity) mean lower product costs.

- **Process optimization.** Besides the direct costs in producing a product – raw materials, labor, power – there are major indirect costs.

  - the inventory of parts and work in progress

  - the labor time spent *not making* products, either because the products were not needed (over capacity) or because the raw materials or subassemblies were not available

  - the cost of the facility to produce the product, especially if it is poorly utilized

  - the cost to store the finished goods until they are purchased

- the cost of lost sales when there were no goods to sell
- the cost of transporting goods to market
- the cost of selling goods

Process optimization minimizes costs by achieving a balance. Having too little inventory may mean wasting production capacity because specialized workers have nothing to do. Minimizing transportation costs may mean building too many plants. Always having enough product in stock to satisfy any order may mean spending too much to keep the inventory.

In each case there is an optimal balance point at which costs are minimized and returns are maximized, although the optimal point may be impossible to predict and subject to change. The optimum can be usually approximated if there is adequate data, and if the data can be easily understood and related to management decisions. That is the value of a good information system.

BUT I'M NOT A COMPUTERNIK

Your reaction up to this point may be "Yes, information is important to me, but we have a whole bunch of people whose job it is to design our computer systems – it's really not my problem." If you believe that then ask yourself how *they* are going to design systems to help *you* do *your* job. Do they understand your job in detail, your objectives, and how you plan to accomplish them?

It's true that in the past a lot of computer systems were built by system specialists who didn't understand the intended use in detail. It's very unlikely their efforts always produced the most effective systems, but it didn't matter too much because computers were very expensive, and were used only for specific tasks; you could afford to build procedures around the computer, if you couldn't build exactly the right system to do the job.

Times are changing. Computers are less and less expensive. Computers are now useful tools for many information needs, not just accounting or complex computations. And there is an overwhelming amount of evidence that says that effective computer systems cannot be designed without extensive cooperation between builder and prospective user.

You may feel that you don't have time to help design a computer system, but will you still feel that way when a lot of money has been spent on a system, and it doesn't work well, but it's too expensive to replace?

We believe that effective computer systems require a team effort between users and technical specialists, but that there is a clear division of labor.

- Prospective system users must specify goals and objectives. These goals must be within the bounds of what is feasible, so the users have to know something about what a computer can do and what is science fiction.

- Technical specialists should make most of the detailed design decisions. But in most cases these "technical" questions cannot be answered without a good perspective on how the system is to be used. Technical specialists must understand the goals and objectives of the system.

So if you want systems to work you need to know something about computers, even if you have no intention of ever writing a program. (But you may change your mind about writing programs because systems are becoming so easy to use that many "programs" are written by the users. It's the simplest way to achieve their purposes.)

MANAGEMENT NEEDS

It may be that your company has an informal way of doing things. Company policy is to hire the right people and let them do their job. If you feel that systematizing your enterprise would be wrong, given the management style, then be very careful of the use of computers.

You may have no use for computers now. *But if you believe that computerized information systems would benefit your enterprise then you have no alternative but to understand how those systems work:*

1. **There is a system.** If the enterprise works today then there must be a system. People don't just arrive at work and independently decide what they will do today. So the question is not whether there should be a system, but whether it will be formally described or even turned into a company policy.

2. **A program is a specific system.** A computer program is a very specific set of rules that specify what is to be done *in all cases.* A program can be written according to a carefully designed specification, or it can be written by a programmer totally on the basis of personal judgment. In either case it will behave precisely as it was programmed to behave. As a tool in an enterprise information system, it will either do the needed function or do some other function. Unless the needs of the enterprise are determined beforehand, by some formal analysis, the chances are very high that the computer system will not do the right job.

3. **Systems have a life of their own.** Once a system is in, chances are it will stay even if it isn't totally satisfactory. After all, it cost lots of money, and it would cost much more to change, and it's probably doing something right. And anyway, why should you believe it will be better the next time?

The alternative is to adopt a systematic approach. Being systematic *doesn't* mean hiring lots of systems analysts, and creating a formal bureaucracy. *Being systematic means being able to write down what really happens in the enterprise.*

The use of computers is helping train a large number of people in some form of systematic thinking. With a little training, understanding systems to the degree needed here isn't that difficult. And there is a most interesting phenomenon that is seen repeatedly when computer systems are added to an enterprise: *much of the benefit comes from understanding what's going on and cleaning it up a little; compared to that the addition of the computer is often secondary!*

DISTRIBUTED COMPUTER SYSTEMS

Information systems aren't new; what's new is the growing number of information systems that utilize computers. The impact of computers is threefold:

1. The addition of computers can make a particular information system more effective by reducing the cost, and by improving the accuracy, timeliness and usability of the information. Computerized systems can deal with more information than can the manual counterparts.

2. Computerized information systems can be accessed from a distance, via electronic commmunications links. This ease and speed of access greatly increases the potential geographical scope of an information system.

3. Computerized information systems can be interconnected with electronic communication links. Interconnected systems can access the data from other systems, as well as the data gathered and managed locally. With many interconnected systems, relationships in the information can be found that would not be visible in separated systems.

Because of their value, computerized information systems are popping up all over. Many different terms are used to describe what they do: "distributed data processing", "distributed computers", "distributed access", etc. All of the approaches have much in common:

1. More computer power is used. As computers become less expensive they are used more widely.

2. Access to the computer system is distributed. In other words, more people access the computers.

3. More of the access is direct and interactive. Traditionally, computers have been utilized indirectly, through agents such as systems analysts, programmers, and keypunch operators.

> As computer equipment becomes less expensive, it is more effective to have the computer system used directly by individuals who generate information (e.g., create a sales order) and individuals who use information (e.g., need to check inventory).

This handbook is about these distributed computer systems, their value, how to plan for them, how to build them, and how to use them.

KEY TERMS

Information. Information is knowledge, something that helps a person accomplish a task. We are concerned with information that can be stored in a computer system, which, in practical terms, means something that can be expressed as text. The information may be traditional computer "data," in the form of arrays of numerical values, or it may be non-numerical information, such as a name.

Enterprise. An enterprise is a group of individuals working together toward some common goals. A business is the most common form of enterprise with respect to distributed computer system use. But the subject is germane to many other enterprises as well – laboratories, manufacturing plants, universities, government agencies. The form and use of information will vary according to the purpose of the enterprise. For a bank much of the information will be accounting and financial data; for a laboratory the data may be experimental results.

System. A system is "a complex unit formed of many often diverse parts subject to a common plan or serving a common purpose" (*Webster's Unabridged).* An information system is a set of functions that work together to make information useful – a mail system, a phone system, or a status report.

Distributed system. A distributed system is one that is spread over an area. For information to be useful at a distance, formal (planned) systems are used. Hence a distributed information system is a mechanism that makes information useful across an area. We will focus on enterprises that are big enough (span enough area), or use enough information, to require formal information systems.

Interactive system. Interaction means dealing directly with the system, rather than through an intermediary. Proximity to information has a large impact on its value. We will be particularly interested in interactive computer systems, in which individuals interact directly with the programs and data they use, with on-line computer terminals.

Computer. We will use "computer" as a generic term, including various forms of data storage and data communication devices, as well as data processors. In general, we are more interested in computers as devices that can store, retrieve, and communicate data than we are interested in computers as devices that can perform arithmetic quickly.

On-line. Computer use is on-line if the response between input (to the computer) and output (from the computer) is rapid, say less than one minute. Data is on-line if it can be accessed via an on-line computer system.

Minicomputer, mainframe computer. We will use these terms to categorize computer systems that cost less or more (respectively) than a certain arbitrarily chosen amount (say, $200,000). Modern minicomputers are not very different from mainframes in the functions they can perform; today it's a question of economies of scale.

Data base and Database. We will use the term "data base" generically, to mean the data stored in a computer system. When we mean to refer to that particular form of computer facility known as a Data Base Management System (such as Digital Equipment Corporation's DBMS-11 for the PDP-11 computer) we will capitalize the term and make it one word – Database. Most of the time we will use the term generically and refer to any kind of data storage mechanism.

Chapter 2
COMPUTERS AS INFORMATION TOOLS

BACKGROUND

The first electronic computers were calculating machines. The term *computer* came from a clerical job classification: a "computer" was a clerk who spent days at a mechanical calculating machine, performing extensive mathematical computations. The Second World War spawned the development of electronic computers that were used in complex war-related tasks, such as calculating the ballistic tables that were used to aim large guns.

The earliest electronic computers were made with vacuum tube circuitry, and therefore were very bulky, required large amounts of power and cooling, and were relatively unreliable. The development of commercial transistors was important for the manufacture of computers because the use of transistors decreased power and cooling requirements, and increased reliability.

The most important technical development was the integrated circuit (IC). An integrated circuit is like a transistor, namely a small element that includes semiconductor material, but instead of having a single active component, like a transistor, an integrated circuit may include hundreds or thousands of components implementing complex functions. Typically, integrated circuits have doubled in complexity each year since they became a commercial reality in the early 1960's. Now complete small computers can be fabricated on a single integrated circuit.

The earliest computers had very little storage capability – typically just a few thousand characters of data. For each program executed, the program and data were loaded into the computer from punched cards or paper tape.

Today even modestly priced minicomputers (less than $100,000 for a total system) may have hundreds of millions of characters of on-line (quickly available) data.

The earliest computers required expensive specialists to code the programs, handle complex operational procedures, and carefully interpret the data.

Today's computers can be used directly by nonspecialists using many kinds of terminal devices, for example terminals that are like typewriters or television sets. The terminals can be located almost anywhere, linked to the computer by communication lines. In the last 25 years, "higher-level" programming languages have made programming more a question of understanding the problem at hand, and less one of understanding the intricacies of the machine. Special "system programs" make computer use much easier by concealing many of the internal details.

Simplistically, a modern computer system is an electromechanical device, which can cost less than $100,000, store hundreds of millions of characters of data on-line and accessible in fractions of a second, perform various processing functions using that data, and transmit data and results to and from remote locations over communication links at rates from 30 characters per second to over 50,000 characters per second.

The technological improvements that have already changed computers to such a degree are continuing today. IC complexity (how much logic or storage can be put on a single "chip") continues to approximately double each year. More complex ICs make computer system central processors less expensive and make it easier to use data storage and facilities more efficiently. Data storage costs (the cost to have a character of data on-line) have been going down by about 35% per year – not quite equal to the rate of IC improvements, but remarkable nonetheless. Data communications becomes less costly as small computers are used to optimize the system (data networks) and with the increasing use of earth satellites instead of cables and microwave links.

The improvements can't go on indefinitely, but the ultimate limits don't seem very close today.

HOW COMPUTERS ARE USED

Computer use has changed along with computer technology. For example, a single IC chip microcomputer today has about the same processing power as the earliest electronic computers. But the IC will be packaged within a piece of equipment, such as a typewriter, whereas the early computer would have been the center of an entire computation department. The computer power is the same; the usage is certainly different.

Understanding how modern computers can be used is important: adding a computer system to a clerical department doesn't mean building an expensive computer room and hiring operators and programmers. To put computer use in some perspective, we will briefly trace the evolution that has occurred.

Early Large Computers

Every early computer was a "large" computer, in the sense that even the smallest of them cost a lot of money. To use a computer, it was necessary to sign up for a block of time, and, when the time arrived, go to the computer room and run the machine by pressing buttons, setting switches, etc.

Batched Job Processing

Training each computer user to run the computer was costly and inefficient. The next step was to add a computer operations staff, and to have the user submit a job such as a deck of punched cards to an operator. The operator would collect several jobs, perhaps sort them on a priority basis, run them as a batch, one after another, and then return the output to the various users. To make the batch processing more efficient, a small system program was added to keep the work flowing – an operating system.

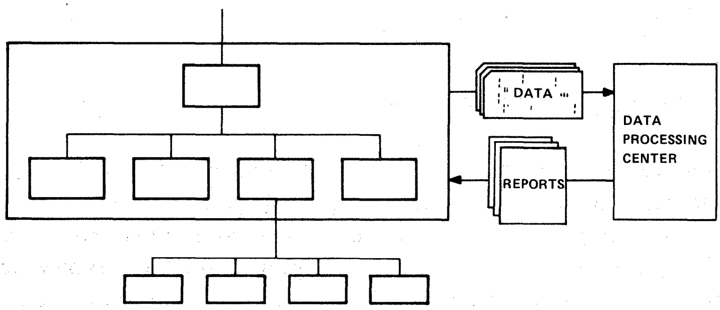


Figure 2-1. Processing needs are met by submitting processing jobs (punched-card decks) to a remote computer center and receiving printed reports.

Multiprogramming Batched Job Processing

As larger on-line disk storage units became available, the batched processing operating systems became more powerful.

- Multiprogramming systems permitted multiple jobs to execute "at once", so that one could be using the central processor while others were waiting for I/O activity to complete.

- On-line file systems permitted data to be left with the system between jobs rather than being reloaded when needed.

- System software was developed for file access; this relieved the application programmer of many of the details. The programmer could specify a file by name, and then read records from the file, without actually knowing where the data was stored, or how to control the various storage devices.

These systems were still primarily used by preparing jobs for batched job entry.

Smaller Computers

Technology improvements permit the building of computers that have more power, but cost the same as bigger, older machines (the general trend in large computers). Or we can build computers that have the same power as previous models, but cost less. The minicomputer and microcomputer both emerged along this constant-power, decreased-cost pathway. The small computers also gave rise to much of the development of interactive styles of computing (a key facet of distributed systems).

Interactive Computing

Interactive computing means that the user once more runs the computer. But now user efficiency is much improved because the computer cost is less, and small operating systems replace the old switch and button manipulation.

The interactive user can correct mistakes immediately, rather than waiting for the return of a batch job some hours later, and can base present work on immediately preceding results.

Small interactive computers are also used for laboratory experiment control, machinery control, and other real-time applications that can not fit into a batched environment.

Timesharing

Timesharing developed as a merging of the multiprogramming, batched job operating systems developed for large machines and the interactive style of computing developed on small machines. In timesharing, many users are connected on-line to a single machine, which is programmed to make it appear as if each user had private use of a smaller machine. The on-line user develops, tests, and executes programs, much as if using the control terminal of a small computer.

Timesharing won immediate acceptance for applications in which the immediate response and turnaround of results had a large payoff, such as small engineering-problem-solving calculations.

The first timesharing systems were designed for programmers, engineers, scientists, and others who had a specialized need for the computer. They had to understand how the system worked.

Then timesharing technology moved into systems that would not be used by computer specialists. For example, bank clerks use a specialized form of timesharing system, called Transaction Processing, to speed up internal accounting procedures.

Transaction Processing Systems

An interactive computer system that is used for data processing applications by people with no computer-specific training is called a "transaction processing" system, simply because the data is processed in single, interactive transactions, rather than in traditional batched fashion.

Transaction processing systems differ from timesharing systems, because the users are different and the nature of the processing is different.

- A timesharing system presents the user with the appearance and capabilities of a complete private computer system, which can be programmed for any purpose. A transaction processing system lets the user talk to a specialized application "engine" – for example, a banking machine. The transaction processing terminal user sees only the details of the application, not internal computer details (such as job control language, file specifications, etc.).

- Most timesharing system use is for small individual problems, whereas transaction processing systems often represent large and complex applications (for example, the entire processing needs of a department). One implication is that the data base on the transaction processing system is much more valuable than the data base on a timesharing system. Given the nature of timesharing systems, it is often adequate to be sure the data base is backed up every few days, whereas transaction processing systems typically include mechanisms to back-up the data every few minutes, in order to minimize the impact of any kind of disruption.

- The timesharing user often has written the program he is using and is typically responsible for everything from validating the input to defining all the processing steps. The transaction application user is just doing a job; data validation and task control must be written into the transaction processing application.

Computer systems that have been specifically designed for transaction processing are an ideal component for many distributed information systems. They solve many of the inherent problems without requiring any additional applications programming or special knowledge on the part of the terminal user.

Query Systems

Transaction processing systems can replace much of the record keeping previously done with paper forms. Once this data is on-line it provides other major benefits. Using a query system to obtain the accurate, detailed information stored in an on-line computer system makes the data useful for many additional purposes. For example, a business manager can explore alternate strategies and make timely decisions. A laboratory director can look for long-term quality control trends. A plant manager can compare the reliability of various pieces of machinery.

Query systems are interactive programming systems designed for extracting data as easily as possible. Rather than having to know the details of programming languages, or the precise formats of data files, the query system user can pose questions of the form "Find me all employee-records for employees with salary less than $10,000."

Query systems, transaction processing systems, and the communication systems that ship data back and forth are the key elements of distributed information systems. With relatively few tools we are able to gather important data from the operation, access it in planned and unplanned ways, and relate it to other data.

COMPUTERS AS INFORMATION TOOLS

There is nothing mysterious about computer-based data systems. Their capabilities can be compared to those of a good filing clerk or secretary handling paper files, with the added benefits that

- a computer system can deal with much larger and more complex collections of data
- a computer system can access the data more quickly
- computer systems can quickly transmit the data to remote users or systems via communications facilities

It's the added benefits that make practical differences and make distributed computer systems such an important topic.

Data Storage and Retrieval

Computers store digital data. This means that data in a computer data base must be represented in text form rather than photographs, drawings, or handwritten notes (although there are means of storing these too). An easy way to visualize a typical data store is as a collection of individual records all written in the same format. For example, each employee record in a file might consist of the employee name, employee number, date of hire, salary or wage rate, department, and the employee number of the supervisor.

Storing large amounts of data is one thing; being able to retrieve specific data is another. For example, suppose that a file consists of 10,000 records, and that in order to find the one that we need, it is necessary to search them in sequence. Even if the search is rapid, it would take quite some time and put a practical limit on the complexity of data files and their use.

Indexed-Access. Fortunately we are able to index data, and retrieve particular records on the basis of the index. For example, with the employee record we might want fast access on the basis of the employee name, or the employee number, or perhaps the department in which the employee works.

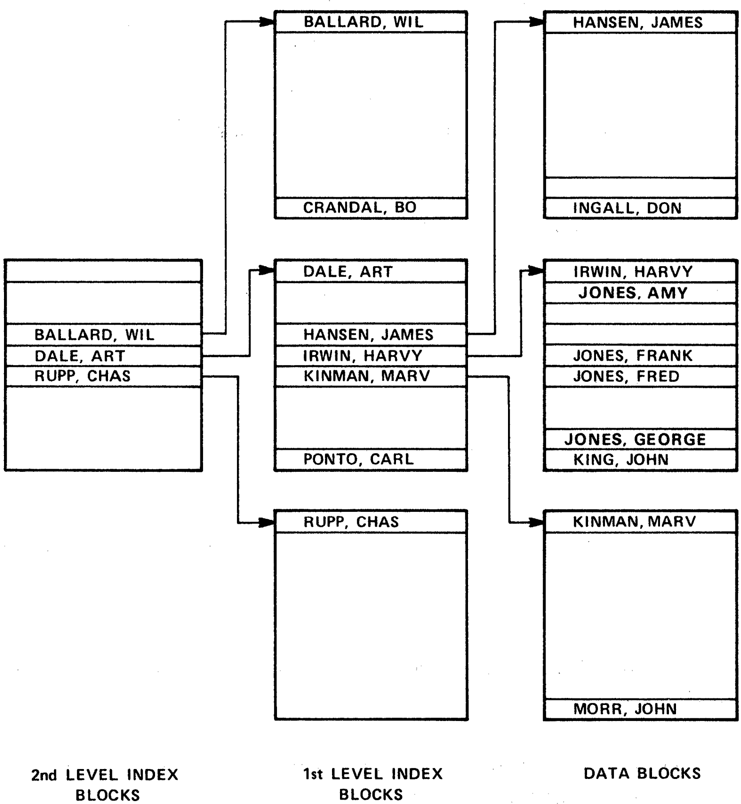


Figure 2-2. Internal file structure of an indexed file. Access to the data blocks is made by using the roadmap provided by the index blocks. Because each index block references many lower-level index blocks (or data blocks), the desired data is found quickly.

To provide quick indexed access to data in a file you add records that serve as a "map" of the data. These records are called **index blocks** and logically take the form of a tree. Suppose we are looking for the data record for employee "Fred Jones":

1. The first index block describes the entire data file at a cursory level of detail. It shows that if we want to find "Jones, Fred" we need to look at the next level of index detail, which covers the alphabetic range "Irwin, Harvy" to "King, John".

2. The detailed index block shows the data block with the records for "Jones, Amy" to "Jones, George".

3. Within the data block we find the records in sequence, and after a brief search, we find "Jones, Fred".

These computer access steps are similar to how we might use a manual filing system:

1. Employee records are in the file cabinet near the windows.

2. Jones, Amy to Jones, George are in the third row, second drawer.

3. Fred Jones must be somewhere in the middle of the drawer.

4. ...

Some points about indexed-access schemes, such as we have been described above:

1. As records are added or deleted we change the roadmap, so the access order is preserved.

2. We can extend the scheme to very large files. For example, when the employee file grows to the point that there are too many first-level index blocks to describe in a single second-level index block, we can add a second second-level index block and provide a third level index to those. Although we

may have many index blocks, the depth of the tree (the number of levels) grows slowly because of the way the tree fans out.

3. We have dramatically reduced the amount of data that has to be searched, since we only need to access the index blocks on the path between the tree "root" and the desired data record, rather than scanning the entire file, greatly improving access speed, as well as providing a powerful, high-level access capability.

Indexed-access is not the only way of structuring data. It is used as an example here because it is a very important method, and by itself provides a powerful information tool.

Data Relationships

Putting data records onto a computer greatly enhances the ability to understand and use the data collection as a whole. For example, if the employee file included data on languages spoken, we could easily search a large number of data records, with a simple query request, to find a Spanish-speaking employee working in the Northeast Region who could act as translator for a customer from Madrid.

Data Communications

The ability to structure and access data is one key to using computers as an information tool. The other key is the ability to provide access to the data from remote locations, or move data from one location to another. Modern systems provide these functions.

We'll discuss data communications in detail later. Simplistically, you can think of data communications as being like using the telephone, except data transmission occurs instead of voice communication, much like a ham radio operator transmitting Morse Code. Setting up data communication is somewhat more difficult than placing a telephone call, but not very much, and it's getting easier all the time.

Electronic data communications makes it possible to access an interactive computer system a thousand miles away as conveniently as if the terminal was located next to the computer. Being able to overcome distance makes a computer information system radically different from a manual information system. With a manual information system you must either go to the data file or transmit your request to a person at the file and count on him to find the right data. With a computer you can stay where you are and still use the files without an intermediary.

Other Benefits

Computer systems provide other benefits besides rapid access to large collections of data, and access from remote locations:

- **24-hour service.** The data can be accessed when it is needed, not just during office hours. This is particularly valuable for geographically dispersed organizations with offices in different time zones.

- **High-accuracy.** When the machine has been programmed correctly it does the right job, tirelessly, patiently, and speedily.

- **Consistency.** A new computer-based record can be verified against existing records (does the account match the name?). To add such checks for a manual system would add a tremendous clerical overhead, further reducing the useful capacity of the filing system.

- **Promptness.** Computer response is prompt, ranging from instantaneous to a minimum delay.

- **Directness.** Manual data files have a tendency to restrict usage, because data retrieval is a slow, tedious, and often dirty process. With a computer-based data file there is no one between the data and the need, and the ink won't run off on hands and clothes.

The Computer as a Policy Agent

A computer system, once programmed to implement a given record-keeping policy, executes that policy with no exceptions. If a copy of each sales order is to be sent to the market analysis office then a copy of each sales order *is* sent to the market analysis office, every time, without exception.

Likewise, any other data keeping and access policies are implemented with total consistency, at all times, and without exception.

SUMMARY

In the previous chapter we introduced the concept of an information system, and showed how any enterprise that grows beyond a certain size needs to set up formal information systems to coordinate and control management activities.

In this chapter we have briefly discussed the evolution of computer systems to where they stand today and how they can be used as effective information tools. We focussed on three types of computer systems:

1. Interactive data processing systems designed to be used by people without computer training or experience (transaction processing systems) are being used increasingly to automate the operational record management of enterprises.

2. Interactive data query systems provide unplanned access to the data gathered by the operations-oriented systems in order to utilize the data for better planning and management.

3. Data communications systems eliminate most of the impact of distance and separation. Data links make the data from an international division as quickly and conveniently accessible as the data from an adjoining plant complex.

A computer, like any other productive tool, must produce benefits greater than its costs. Any information system has both direct and indirect costs and benefits. Let's see the changes that can result from adding a computer to the system.

- **Direct costs.** Adding a computer system adds cost, obviously. But a computer-based information system may well displace more than its cost by reducing the clerical staff that is needed, or by permitting the same staff to do more cost-effective work. The cost of computer technology continues to go down while the cost of labor rises; computer-based information systems become increasingly cost-effective.

- **Indirect costs.** A well-designed computer system can have many indirect benefits as well. Individuals that don't use the system directly may find their work expedited as well. Customers may find the business more responsive to their needs and problems, and do more business.

- **Accuracy.** A well-designed computer system is certain to raise the accuracy of the data record system.

- **Timeliness.** Similarly, a well-designed computer system is certain to provide more timely access to data, once the data collection grows beyond a certain point.

- **Proximity.** A well-designed computer system can provide data access directly to the person needing the data.

- **Size Limits.** Not only can computer-based information systems replace many manual information systems, but computer-based systems can be extended to much larger total size than would be feasible for a manual system. Many of the large computer-based systems today could never have existed on a manually kept basis. The use of computers adds a new dimension to record keeping, and has played a substantial part in the creation and management of large conglomerate or multinational corporations by permitting some degree of common data management.

- **Value of Data.** Also, as we discussed above, computer-based information systems permit the data to be used in new and valuable ways, since the relationships in large collections of data can be found and used.

Chapter 3
EFFECTIVE INFORMATION SYSTEMS

We have briefly discussed information systems, and how computers can be useful tools for building those systems. Now we focus on some of the key factors in building effective computer-based information systems.

- The value of planning
- Designing systems for organizations
- Information and organizations
- Local information needs
- Long term planning

PLANNING

In all aspects of computer system development, the value of planning cannot be overemphasized. Planning doesn't mean that the fine details have to be worked out at the beginning and then adhered to slavishly. Planning does mean having a pretty good idea about what you are trying to accomplish and how you can tell if you are succeeding. In fact, the biggest value in planning is that checking results against plans shows where the problems are, so you can find solutions.

Planning has particular importance in computer systems:

1. Computer systems can be much more complex than the manual systems that people have dealt with before. Informal, back-of-the-envelope planning, which may have worked well for simple manual systems, may fail when applied to computer systems, just as informal planning is less adequate as an organization grows in size.

2. Computer systems undoubtedly acquire an existence of their own. For example, programming has a cumulative aspect – after a few years you have much more investment in programming than you could afford to spend over any short period. If the computer system performs some useful function, you will probably keep it around because it would cost so much to change. It's much better to build the right system in the first place than to try to fix it later on.

The best plans are simple and complete. A complete plan that can be summarized on, say, 5 pages, is better than a sloppy plan that's 100 pages long. If the plan is all in 5 pages it's hard to miss understanding what's important and what's secondary. Major holes will show up clearly in a 5-page plan, but may be totally obscured in a long plan.

BUILD SYSTEMS AROUND ORGANIZATIONS

Make the information system fit the needs of the organization; don't design the organization around the limitations of the information system. That principle seems obvious, but it hasn't always been followed in the past.

Over the years, the prices of computers have decreased, to the point that today's computer cost is only a small percentage of the cost of an equivalent computer 10 years ago. When computers were more expensive, they were necessarily a focal point: rooms were built around them, and people trained to run them and use them. These additional costs were small compared to the cost of The Computer.

But technological progress has changed all this.

1. Computers have greatly decreased in cost.

2. More people are using computers.

3. People are using computers as part of their job.

In fact, the economics of the total information system, including both computers and computer users, has changed a lot. The cost of the computer equipment is still significant but much less so than formerly. We have less need to optimize the investment in computer hardware, and more need to make the computer system optimally easy to use.

For example, suppose that we have a large minicomputer system with a price of $200,000. Let's say it costs us $10,000 a month to run the system hardware, including paying off the purchase. Let's assume there are 40 on-line users, for whom the computer is an important part of their job. Let's assume an average salary (including benefits) of $15,000 per year. That means that we are spending about $50,000 a month for the users, ten times what we spend for the computer system. We can afford to double our investment in hardware to $20,000 a month if we can make a 20% improvement in the productivity of the system users.

Optimizing total system costs is more difficult than just optimizing computer system operation. It was relatively comfortable to know the cost problem was the computer and to hire a good data processing manager to make sure the money was well spent.

But the computer is less and less likely to be the problem in the future. And you can't hire a data processing manager to optimize the whole operation – that's a general management job already.

The solution lies in managers' realizing that the computer is their tool and their responsibility. If management wants to leave the design of critical information systems to specialists, and to vendors' representatives, they can't complain about the results. Computers are no longer special tools to be handled by specialists; they are general tools that will have profound effects, good or bad, on the enterprises using them.

The key point to remember is that the computer has to help the users if the system is to work well; it's no longer reasonable to organize the work around the computer.

INFORMATION FLOW WITHIN ORGANIZATIONS

The general shape of organization charts, representing the way large groups of people divide authority and responsibility, has remained remarkably similar since the time of the Roman Army. Either organization builders have displayed a remarkable lack of imagination, or these structures represent fundamental human attributes. We'll assume the latter.

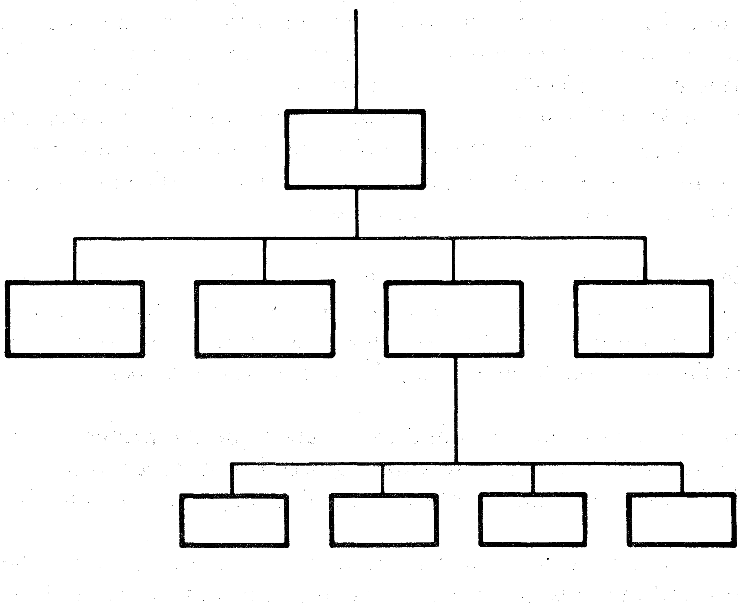


Figure 3-1. A large organization is divided into a hierarchy of departments, with a well-defined chain-of-command.

The structure of organizations reflects man's limitations in dealing with complexity. Rather than having all supervisors and foremen reporting directly to the boss, we have an intermediate level of middle managers. Typical structures have 5-10 individuals reporting to a manager. The idea is that one person can deal with up to, perhaps, 10 sets of issues, but not many more. Instead of having 100 supervisors reporting to one boss, we have 10 or 15 middle managers receiving reports from the supervisors and abstracting the reports to fit the boss's need for information.

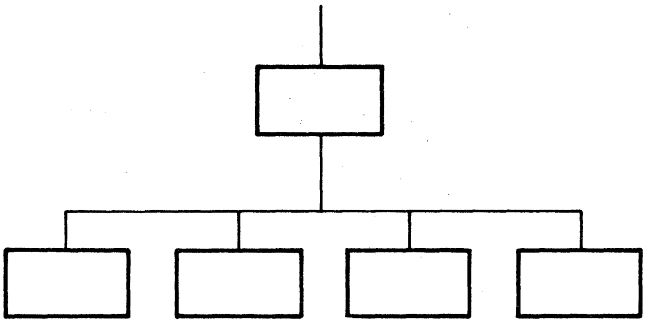


Figure 3-2. Only a small number of individuals directly report to each department manager. The manager can focus on a small set of problems.

In a properly functioning organization, each intermediate person in the structure provides an intelligent information processing function. Report data that flows up to a manager is not just stuffed in an envelope and passed upward. Instead the manager abstracts the data, and tries to form a picture of the whole operation below him, focussing on key accomplishments and problems, and then passes this summary data upward.

Similarly, when decisions are made and passed downward, it is the responsibility of subordinate managers to interpret these decisions and expand them into operational detail.

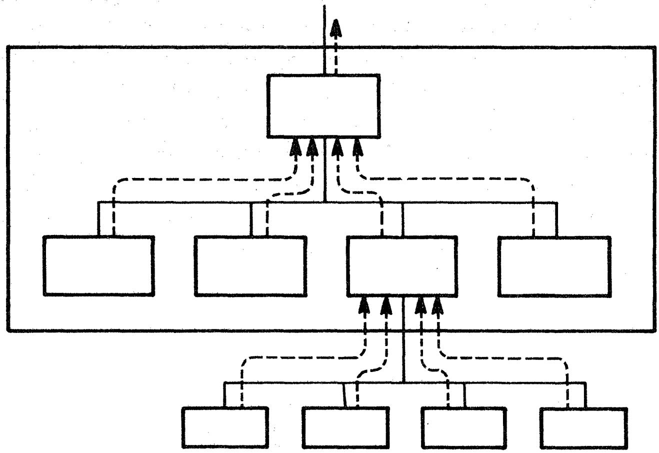


Figure 3-3. As reporting information flows upward in the organization, each department manager performs intelligent information processing, abstracting key items and themes to pass on.

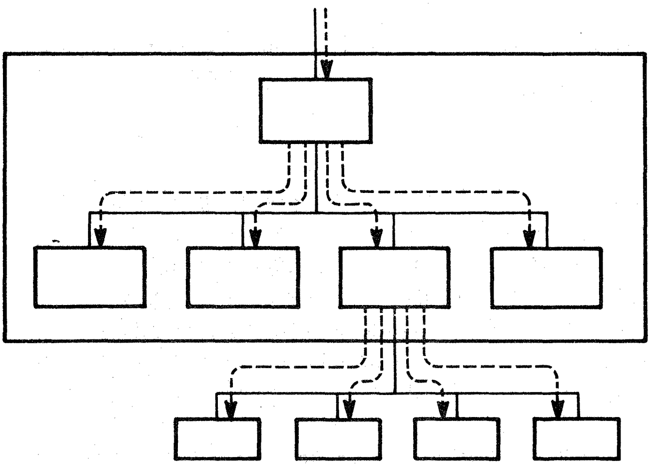


Figure 3-4. Management control information is expanded and detailed as it passes downward in the organization.

Some of the early attempts to build Management Information Systems (MIS) failed because they didn't pay enough attention to the importance of the information processing performed by middle-managers. The designers failed to understand how little value there was in the fact that the President could find out how many cartons of ping-pong balls were on a specific truck leaving the loading dock. The data was of no value, because it hadn't been abstracted to the level where it had meaning (trends in the accounts receivable, inventory quantities, shipments versus projections, etc.).

INFORMATION USAGE

Most information is used locally. Of all the information used in a department

- 80% was *generated within* the department, and only 20% *came from outside;*
- 80% will only be *used within* the department, and only 20% will ever be *sent outside* the department.

This is a natural result of the decreasing value of information as one gets further from it.

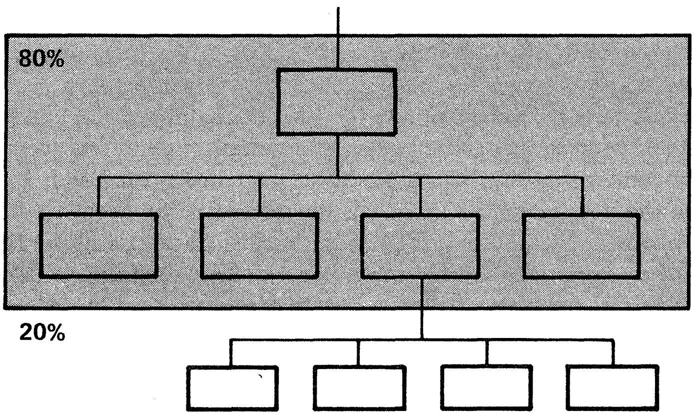


Figure 3-5. Most of the information used locally within a department is generated in the department. Most of the data generated in the department is used only within the department.

One reason that most information is used locally is that much of the information is usable only locally. For example, an accounts receivable department may develop an ACCOUNT AGING factor, which represents the importance of collecting from that account. Everyone in the department will know about this data and what it means. But outside the department, will people know whether a high ACCOUNT AGING factor is good or bad? How much worse is 30 than 15? In other words, although it may be possible to electronically transfer data all over the world in a split second, the data may have little value because its meaning is only locally understood.

There is no simple solution to this problem. If you require each department to meticulously document each data item it uses, all productive work will come to a halt. If a data item isn't fully documented, then the value is private. You end up with a balance, such as we have today, in which only a fraction of the data is carefully documented. This fraction represents the data received from the outside or sent to the outside. The value of documenting data usage is that it can be used for wider purposes; the disadvantage is that it becomes more rigid and constrained, since more parties must cooperate if it is to be redefined or refined.

LONG-TERM PLANNING

It's very easy to underestimate the long-term impact of a computer system, or even a specific program. Computer decisions seem less important when they are made than later down the road. By the time the new system is installed, the users trained, and the minor problems fixed, you're a lot less enthusiastic about making major changes. Even if you fully intend to change something, when the time arrives for planned changes there are more important things to do, and they take priority.

The right philosophy is to believe that every development is permanent, unless the replacement or termination date has been specified, and the costs to replace or shut down have been firmly budgeted.

Just as systems that were intended to be temporary turn out to be permanent, systems that were to last forever soon need alterations. The need for change comes from many sources:

1. **The enterprise changes its goals or methods.** The world itself is hardly static. A major value of computer systems is that they permit managers to react to the need for change. Change in an enterprise can easily create the need for change in the information systems serving the enterprise.

2. **The information system changes the way work is done.** Computer systems can be very powerful tools. As in the case of other tools, they can change the way that work is done. A change in work pattern may make the information system design less than ideal and require it to change.

3. **The initial design may have been wrong.** One proven way of making good computer systems is to install a system and then carefully analyze what's wrong with it. It's easy to get concrete design help from prospective system users if you give them a system to use that they don't like; they'll be eager to tell you why they don't like it. It's not the ideal way to build systems, but it may be one of the more effective ways.

4. **The initial technology becomes obsolete.** Any system design will become outdated eventually, as long as the rapid technological progress in computers continues. Trade-offs that are reasonable when the system is built may become quite unreasonable if the cost of computer hardware goes down by a factor of 2 or of 10.

Systems that were designed with the assumption that they would have to change later are usually much more changeable (and changeable at a lower cost) than systems that were built to last for eternity. Planning for change is a good bet.

SUMMARY

If there are three keys to building effective computer-based information systems, they are

1. Understand the organization of the enterprise for which the system is being designed.

2. Design an information system that fits the organization.

3. Emphasize high-quality, ongoing planning.

Chapter 4
MINICOMPUTERS AND INFORMATION SYSTEMS

WHAT IS A MINICOMPUTER?

A "minicomputer" could be defined as a computer with a certain performance, but that definition changes often with the rapidly improving cost/performance of all computers. Instead, we define a minicomputer simply as a computer system within a price range of approximately $20K to $200K.

A minicomputer has a fairly constant characteristic cost, requires a certain amount of additional investment to make it work, and requires a certain amount of operational support. While these factors remain relatively fixed, the magnitude of work a minicomputer system can perform continues to increase.

A modern minicomputer, such as the Digital Equipment Corporation VAX-11/780, is very sophisticated – comparable in most respects to a "mainframe" (system costing more than $200K). For example, the VAX-11/780 has virtual memory, demand paging, and machine instructions for decimal arithmetic, character string processing and editing, and high-performance calculation: previously mainframe characteristics.

In the past many processing tasks done on mainframes were unsuitable for minicomputers, because of the minicomputers' minifunctionality. Today the decision between mainframe and minicomputer can be made primarily on the basis of scale (how much system power do you need), and system engineering (where do you need the processing and data).

It is important to keep in mind that modern minicomputers have all the key system functions, such as file systems, indexed-access record management systems, query languages, and full Database management systems.

MINICOMPUTER USAGE

Minicomputer operating systems are designed for particular processing functions, not for large general purpose data processing centers. Minicomputers traditionally have been used for interactive computer applications. The advantages of a functional system design that focusses on doing a specific job (for example, an insurance company's Workman's Compensation Claim Processing) are many.

- A simple, efficient and effective operating system
- Cost savings
- Responsiveness to local needs

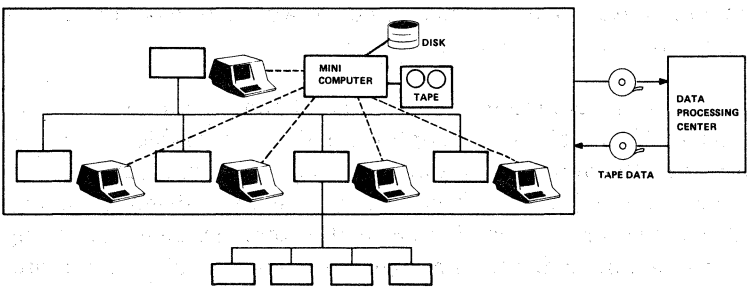


Figure 4-1. A local minicomputer serves the need of a specific department. In this example, data is exchanged with a large computer center in the form of magnetic tape reels.

Let us take a further look at these advantages.

Simplicity of Operation

Many functional systems don't need any highly-trained operational staff at all. If the computer system performs a specific application, then all computer operations can be designed to fit into the application. That means anyone who is trained in the application (in our example, someone who understands Workman's Comp) can run the computer system with little additional training. The system operator (the person in charge of machine operation) could well be the Workman's Comp clerical supervisor.

Since the system serves only the needs of the department, and is not used for program development, any extraordinary processing requirements (the kind of thing that requires special operator attention) will be caused only by the natural growth of work in the department. Highly-trained operators are not necessary.

Minicomputer Cost Savings

In terms of lowest cost per instruction executed, computer system cost/performance improves with increased computer size – a simple economy of scale. Of course, the workload must be large enough to warrant using a large machine, and all the work must be done at a single processing location.

Another set of cost savings is inherent in the use of smaller computers. If an application is designed to use multiple smaller computers, rather than a single, large computer, the following benefits can be realized.

- **Low initial cost.** Purchasing a minicomputer system is much less costly than leasing an underutilized mainframe, sized for full system capacity. If an application is designed so that it can expand beyond a single system (see following), hardware costs can be minimized by initially buying only enough for application development and pilot test, and then expanding later.

- **Incremental expansion.** As more system power is needed, it can be added in relatively small portions, rather than in large capital outlays. In other words, a minicomputer-based system can be expanded in small, less costly steps, and the amount of computer paid for kept more tightly aligned with the amount of computer actually used.

- **Lower cost expansion.** Additions to a minicomputer-based system can be made using the most modern technology. Often expansions to an existing large system must be made with the same technology with which the system was made. Since technology is improving rapidly, the modern technology will often be more cost-effective.

- **Minimized communication costs.** Processors and data can be located where they are needed. Just as careful placement of factories and warehouses can decrease product costs by minimizing transportation, carefully locating data and processors decreases information system costs by minimizing the cost of data transportation.

- **Responsiveness to local needs.** A functional system serves a well-defined purpose designed specifically for its local users, whereas a bigger system may serve many independent functions. Therefore it's easier to adapt and alter a functional system, since there is less negotiation and since priorities are easier to establish.

NETWORKS OF MINICOMPUTERS

The obvious problem of building computer applications with minicomputers is that the application may not conveniently break into many independent pieces. Some applications are ideal. For example, if a minicomputer system is designed to run a gasoline station, then you add a new system when you add a new station. None of the systems is particularly concerned with how many systems may be doing the same function.

But because most information systems require some cooperation and communication between the functional pieces, computer net-

Table 4-1. Minicomputers and Mainframes

	Minicomputers	Mainframes
Typical system price	20K–200K	200K–
Highly-trained operators	few or none	full staff
Systems programmers	few or none	often
Design specialists	no	often
Adaptation of technology	rapid	slower, to protect lease-base income
Usage style	interactive timesharing and transaction processing	often batch primarily
System priorities	factored into design	established by operations staff
System changes	easy to make	complex and often difficult
Architecture and instructions	comparable	comparable
Software functions available	comparable	comparable

working technology has been developed. Computer networks interconnect multiple computer systems by using electronic communication links such as telephone lines, so that the systems can work together effectively:

- Programs on different systems can send messages to each other to advise on progress and problems.

- A program on one system can use data stored on another.

- A smaller system can use expensive peripheral devices on another system when needed. For example, many small departmental systems might share a single Computer-Output Microfilmer (COM).

- Remote systems may even be operated remotely, with program loading and execution commanded over the network communication links.

Networks of minicomputers can't replace an arbitrary large computer application running on a single system. Effective multicomputer systems require that most local processing requirements are supported on a local system, and that the network is used relatively infrequently, compared with access to local data. Fortunately most information systems are characterized by local data use – most of the data used in one place is generated there as well – and hence can be designed to use smaller computers, and reap the benefits mentioned.

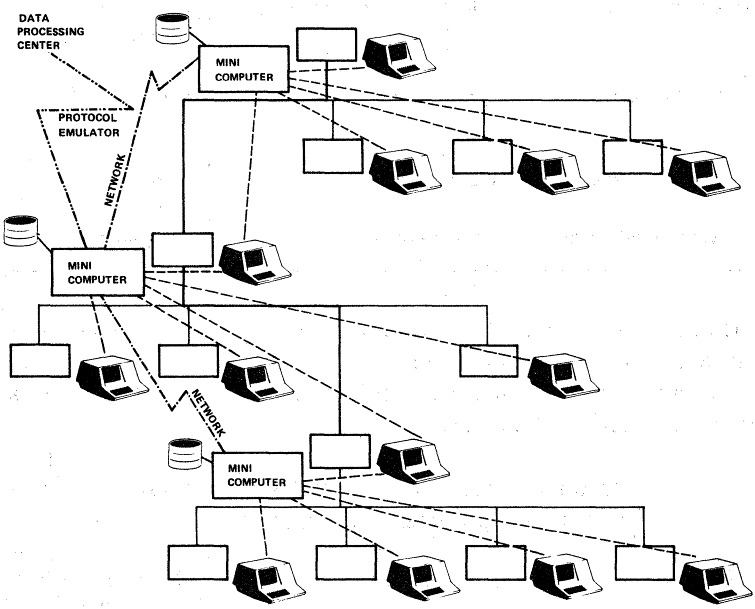


Figure 4-2. Many departments use local, functional minicomputers. The minicomputers are interconnected into a data network. The network connects to a large mainframe.

Computer networks also make multiple minicomputer systems easier to manage. Most enterprises don't want each department to set up its own data processing center, with different machines, local programming standards, and incompatible ways of doing things. If the different systems are connected into a network, they are much easier to manage as a whole. For example, program design, implementation and maintenance can be done from a central group, for major application development.

MINICOMPUTERS WORKING WITH BIGGER SYSTEMS

Most larger enterprises will continue to use large computer systems for intrinsically centralized processing, such as corporate accounting. Large computers provide economies of scale.

Even where large, centralized processing centers continue to exist minicomputers can be a most valuable tool.

1. Minicomputers can lower the cost and maximize the responsiveness of access to a large computer system. Although they may not be called minicomputers, all large interactive computer systems use minicomputer systems internally as communications controllers, communication network processors, intelligent terminals, and intelligent terminal controllers.

2. Smaller, functional systems can be linked into the large central system to provide rapid and convenient communication. For example, day- and week-end roll-up summaries can be sent from the functional system to the processing center electronically; corporate reports and data generated by the large system can be sent automatically to the functional system, to be used locally.

SUMMARY

Minicomputers can be characterized as computer systems of a fairly constant price that have traditionally been used for specific processing functions, often interactively. The power of a minicomputer has changed so that modern minicomputers have the logical capability of mainframes, and can be powerful information tools. By interconnecting minicomputers into networks, many large applications needs can be fulfilled effectively.

Chapter 5
BUILDING DISTRIBUTED SYSTEMS

As computer cost/performance improves, more and more enterprises will acquire computer-based information systems. Because information is a subtle and valuable resource, in general, the design of these systems cannot be left totally to outsiders (such as computer system vendors, and systems houses). Only members of the enterprise fully understand the information needed for daily operation. Only the management of the enterprise fully understands the long-term goals and objectives of the enterprise.

As modern computer-based information systems become increasingly valuable to management, they require more management attention than in the past for many reasons.

- A greater percentage of individuals in the enterprise are using a computer system directly; thus increasing the importance of making sure that the system does the right job.

- The computer system can do more than just specialized functions, such as accounting; thus making it more difficult to leave the design to specialists.

- A computerized information system implements many policies that have traditionally been considered management questions – Who can access particular data? Where are decisions to be made?

- A computer-based information system is potentially one of a manager's most valuable tools. Managing means responding to problems and to changing requirements. A computer system puts into the hands of a manager more information with which to make better decisions.

Although managers and other computer system users must take a more active role in the design and implementation of computer systems, sensible division of labor dictates that there still be computer specialists:

- **EDP (Electronic Data Processing) and Information System Managers.** Computer-based information systems will be an essential resource for the enterprise, and will justify high-level, specialized managers.

- **System Designers.** Designing high-performance, usable and cost-effective computer systems requires trained design specialists. Depending on the enterprise, the system designers may or may not be the system managers.

- **Programmers.** Implementing computer systems requires programmers. In some cases the programmers will also be the system designers. To a growing degree, the system users themselves will do some of the application "programming", by using simple, interactive utilities, or problem-oriented "programming" systems.

A distributed computer system is designed and built in the following phases:

1. The enterprise is organized for effective division of tasks and responsibilities.

2. Ways are devised by which individuals interact with the computer system. To a growing degree, interaction is direct, by the use of on-line terminals.

3. The computer functions to be implemented are aggregated into what we will call *applications*. The reason for having individual applications, rather than just one big system, is similar to the reason for dividing an enterprise into departments in the first place. We want to divide the information system into smaller, less complex, more manageable subsystems.

4. A basic strategy is devised for implementing the system. This includes choosing the type of equipment and communication facilities to be used, and the method by which the system development will be managed.

5. The system is designed in detail by technical specialists.

6. The system is programmed.

In reality, this isn't a one-shot process – reorganization, redesign, and new implementation will continue as long as the enterprise or the computer technology changes. But these phases help explain how the distributed system works together.

- General managers are responsible for the overall operation and direction of the enterprise. This must be reflected in the design of the system. The system must implement the enterprise decision-making and data access philosophy, for example. General managers must make their information-related policies and goals clear, so that they become a basic part of the computer system design.

- EDP managers must set overall computer system design and operation policies so that the general management direction can be followed, and a reliable and cost-effective computer system can be built.

- Individuals for whom the computer system will be a primary tool with which they do their jobs must participate in the design of the application.

- System designers work within the policies and plans set forth by EDP management to develop applications that meet users needs.
- Programmers build these systems.

The next eight chapters examine in greater detail the process of building distributed computer systems:

- **Chapter 6 – Key Management Decisions** examines the role of the general manager.
- **Chapter 7 – Key EDP Management Decisions** discusses the responsibility of EDP managers.
- **Chapter 8 – Usable Systems** looks at the question of the productivity of the system user, and what design decisions most affect it.
- **Chapter 9 – Information Work Stations, A Design Model** describes a problem-solving technique in which an information system can be partitioned into specific applications.
- **Chapter 10 – Data Communications and Networks** looks at some of the details of modern data communication technology (some basic ideas were introduced in Chapter 4).
- **Chapter 11 – Technological Forces** describes computer technology evolution briefly. The topics are chosen to help an EDP manager make good long-term planning decisions, and to give system designers some help in adapting design strategies to keep up with technology.
- **Chapter 12 – Distributed System Design Goals** discusses some of the detailed design areas and goals intended to make systems more usable.
- **Chapter 13 – Distributed System Design Techniques** provides solution approaches to many of the problems involved in achieving design goals.

SUMMARY

Managers and key system users must be involved in setting distributed system direction and making sure the system will work within the management and organization philosophy of the enterprise. Managers, users, and computer specialists form a team that must work together if effective systems are to exist. Subsequent chapters go into greater detail on specific tasks and problems that are encountered in distributed systems.

Chapter 6
KEY MANAGEMENT DECISIONS

Managing an enterprise is coordinating the work of many people toward some common goals. Much of management is information processing: disseminating information outward through the organization to give direction; gathering information from operations to evaluate progress; gathering and evaluating information from outside the enterprise. A good manager deals effectively with informal information (person-to-person skills), and establishes and uses formal information channels (regular status reporting, various forms of performance evaluation, etc.).

The use of computers can decrease the cost of processing information, increase the value of information, and facilitate the construction of larger total information systems. Good information systems can provide competitive value by reducing both management and production costs and by permitting the enterprise to respond to changing marketplace needs more rapidly.

An effective general manager must perform several tasks:

- Set fundamental direction for the enterprise; this consists of basic goals and strategies for achieving those goals;

- Develop an organization capable of accomplishing the objectives of the enterprise;

- Develop an organizational style that sets local models for the delegation of authority and responsibility, the sharing of information, and like subjects.

- Observe the performance of the organization against the goals and objectives and provide constructive guidance.

In the past, computing, or data processing, was of secondary importance to most general managers. The typical general manager left the computer operation to a department staffed by computer specialists. If that approach was not ideal, it was reasonable:

1. Initially computers were quite expensive, and were applied only to focussed tasks, such as accounting, with limited general impact on the enterprise.

2. The early computer systems had limited capabilities and were difficult to use; much translation occurred between application need and computer implementation.

Now, with computer capabilities both expanded and simplified, an effective manager can no longer leave data processing to the specialists! The division of labor necessary in the past (manager/computer specialist) is no longer required, nor is it adequate to solve the problems at hand.

1. Computers are no longer expensive – computers that might have been corporate systems two decades ago are now sold as toys for home use. In the future, most individuals will use computer systems in their work. There is no longer an obvious partition between the problems of the general manager and the applications that are computerized.

2. The increased power of computers has shifted the focus from finding the optimal implementation of a given application (definitely the job of a computer specialist) to finding the means for implementing applications at the lowest possible cost (without computer specialists).

   Adding the services of a top-quality programmer increases the costs of a computer application substantially. Every computer vendor realizes this, and is investing in making computers usable without special training or skills. In other words, the most cost-attractive computer system possible is one that can be "programmed" and used by the application end-user directly, without professional programmers and analysts.

THE MANAGER'S ROLE IN BUILDING INFORMATION SYSTEMS

A general manager must be concerned with information systems because information is an important asset for managing, and because any extensive information system will implement management policy, by delegating problem solving, permitting and restricting access to information, etc.

Just as a general manager can drive safely without understanding how to forge the pistons in a car, so the same manager can direct an information system development without understanding computer technology. In most cases a general manager can't be involved in the technical minutiae and still spend an adequate amount of time managing.

Although a general manager cannot control information system development in detail, its management is straightforward once the fundamentals of computer systems are understood. The management process for information system development is similar to the management of countless other activities:

1. **Set goals.** An information system is the nerve system of the organization and must be designed to support the goals of the organization. In turn, those goals must be clearly defined and explained to the system builders.

2. **Require planning.** Planning is key to effective computer systems. Without careful planning the resulting system

   - is unlikely to provide the right service
   - may cost too much to throw out entirely, and thus become a managerial millstone for many years while it slowly evolves toward what is really needed

Computer system plans should be summarized in a few pages. If system plans can't be tersely summarized, so the manager clearly sees how the system will work, then it is unlikely the system will implement the right function.

System planning for data processing purposes often provides unexpected benefits by exposing previously unrecognized management problems and opportunities.

3. **Manage to the plans.** Use the plans. Develop a means of tracking progress against them. Often this means that the plans should include crisp and easily tested milestones of accomplishment. Change the plans where necessary, but don't discard them.

4. **Reassess goals periodically.** As the enterprise grows and evolves, and as the outside environment changes, management goals and strategies will evolve. Some changes in management may require changes in the information system. Install a process for periodically reviewing overall information system goals and strategies, updating the system to keep pace with the needs.

DISTRIBUTED SYSTEM CONCERNS

The process presented above by which a general manager can constructively guide information systems is independent of whether the system is local to one department, or geographically distributed.

When an information system serves and links separated departments, however new concerns arise:

- differentiating local need from general need
- planning for major organizational change
- protecting data security

Local Needs and Global Needs

Information is primarily used locally – most of the information in a departmental system has value only to the department. Yet much of the potential of modern, distributed information systems comes from being able to move data around a large organization quickly and accurately.

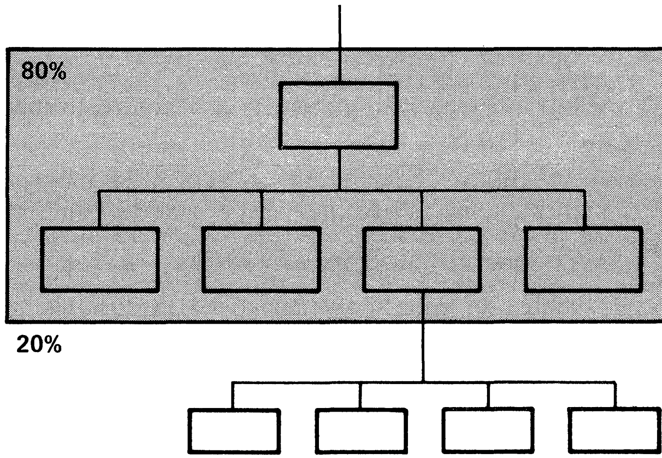


Figure 6-1. Most information is particularly valuable in one location. Eighty percent of the information used in each department has little value outside the department.

Deciding how much data should be private and how much global is a critical judgment in which the general manager must often participate. The control of data is inseparable from the question of who controls the information system. The arguments for keeping some data and control local include the following:

1. Local data can be managed more informally, with less control and procedure, and therefore has more value to the local users.

2. Having a locally controlled information system contributes to the spirit and productivity of the department.

3. Some partitioning of the total information system is usually necessary to make the complexity manageable.

4. Making too much data globally available will lead to counterproductive uses of the information, such as a middle manager manipulating internal accounting schemes to brighten his own image in interdivisional profitability competition.

The arguments for having more global data and control include the following:

1. The maximum benefit from data can be achieved if it is available to all.

2. Local control of data and systems leads to local computer system empire building, which ultimately causes control of the overall system to disappear along with its usefulness.

3. Excessive information privacy is counterproductive to establishing the goals of the enterprise as a whole.

In most cases, the right balance is somewhere in between. Where the boundary should be is a question of organizational and managerial style.

Planning for Change in the Organization

Installing computer-based information systems makes it potentially more expensive to change the organization structure of the enterprise. Changes in the organization that alter decision making or move work functions between departments will require some adjustments in the information system. Depending on how the system was initially planned and built, these changes can be either simple or very costly.

There are three areas of concern:

1. **New Hardware.** Some changes could require changes in computer equipment. For example, if a department were reorganized into two geographically separated departments, then the single computer system that served them before might have to be replicated, or communications equipment added so that one department could use the system remotely.

2. **Reprogramming.** Programs don't wear out and are not necessarily obsoleted by advances in technology. Most information systems evolve to a point where the programming budget is used primarily for ongoing program maintenance, not new development or replacing existing programs. But a change in organization can require a massive reinvestment in programming.

3. **Retraining.** There are hidden costs involved in changing the information system, such as retraining the system users. Even if the users do not require formal retraining, a substantial loss of productivity may occur during a changeover.

Computer systems can be designed to anticipate the need to change, in order to reduce costs when the change is made. Conversely, systems that are not designed to anticipate change usually don't change well. To the degree that a growth and evolution philosophy exists, it should be a factor in information system planning. Because of the additional costs of changing an information system, anticipated growth and evolution should receive conscious management attention.

Protecting Data Security

With a good information system a manager can conveniently work with large collections of data from many geographically remote locations. It should be easy to wander through this data base, exploring hypotheses and discovering the essential nature of enterprise activities.

On the other hand, with a good information system, a malevolent person may conveniently use the data for purposes that are totally opposed to the goals of the enterprise. Unfortunately, there is no easy solution to this problem. The wrongdoer needn't be an industrial spy or anarchist from the outside, but merely a disgruntled employee looking for a stepping stone to a better job on the outside.

The outlook is not totally bleak: most of the information in a system has relatively little value outside the enterprise. But the general manager should be sensitive to the security implications:

1. Having data electronically accessible really changes the nature of security; it's much different from storing notes in a file cabinet.

2. There are no perfect technical solutions; no computer system can solve the problem of the "inside job", or even protect fully from a clever outside penetrator.

3. Internal enterprise politics may lead to misuse of data. Pressure to perform may tempt productive individuals to violate intended data security.

Information security should be addressed in plans, and the goals and accomplishments periodically reviewed.

BUILDING THE RIGHT SYSTEM

Do you know how information flows in your organization? Most managers don't, and for good reason: they have delegated many of the decisions on how to implement processes, and aren't concerned with the details.

But if you don't know how information flows, how are you to build a supportive computer system? Let's start with a couple of critically important DON'Ts:

1. Don't use existing organization charts or procedure books. They weren't intended to serve this purpose. They focus on

critical problems, like division of responsibility, but they probably weren't intended to serve as an encyclopedic reference.

2. Don't build what any single individual, including yourself, *thinks* the right system *should be*. Even if you think that you are running the organization in detail, the chances are that the smart people working for you have their own ways of adapting your directions so they will work more effectively. Consult carefully with system users; they will have valuable suggestions for improvement.

As for DOs, *find out what really happens*. Spend the time, effort and money to discover the real information processes in some detail before plunging into computerization. In addition to building better computer systems, you may well find that a major benefit of the whole effort has been getting a better systematic understanding how your enterprise works!

THE BENEFIT OF YOUR PRESENCE

We've discussed managers' involvement in data processing system plans to ensure that the system meets the goals and needs of the enterprise. Having key managers involved also makes a big difference to the people who work with the information. If, for any reason, the majority of a computer-based information system users dislike the system, use of the system will not be productive, and the system will fail. One major study showed top management involvement to be the key differentiating factor between success and failure.

Why might this be true? For one, putting in a computer system can be a major change in the way work is done. People naturally fear change, especially when it comes in the form of computers, which are viewed as mysterious and unknowable. If you don't know what a computer can and can't do, it's hard to suppress rumors such as "computer installation will take our jobs away!"

At a higher level in an organization, the addition of a computer system may be viewed as part of a power play by one department – "Why should I cooperate so that *they* will look better?"

In all cases, it is a tremendous benefit if key managers are visibly involved in planning the information system, and can provide understandable answers to questions concerning it.

SUMMARY

General managers must become actively involved in planning for computer-based information systems. The system is a valuable managerial tool, and an implicit managerial policy agent. Computer system development benefits from careful management more than most activities. Computerizing an information system increases the need to plan for organizational growth and evolution, and requires a more careful examination of data security. Finally, studies have shown that key management involvement is a critical factor in determining whether an organization will accept or reject a computer system.

Chapter 7
KEY EDP MANAGEMENT DECISIONS

The prime responsibility of an Electronic Data Processing (EDP) manager is the management of one or more computer systems. Some enterprises that use computers have no EDP managers, under that title. For example, in a small company using a small business computer, computer system management may be a minor portion of one person's total responsibility. Larger organizations may have more than one EDP manager by this definition. Any enterprise big enough to think about distributed information systems will probably have at least one EDP manager.

In the past, many EDP managers ran fairly closed shops. Their primary focus was on keeping costs under control while delivering reliable services. It didn't make too much difference which jobs were run as long as the computer was paid for, and growth could be anticipated early enough to order additional capacity as needed.

The role of the EDP manager will have to change, if effective systems are to be developed in the future.

- As the systems become interactive, and as more individuals use the system directly (rather than indirectly through a systems staff), the design of the computer system *must* reflect the very basic design of the enterprise. It is no longer just a question of which vendor should provide *the* system, and how big it should be, but also questions like do the many systems work effectively together, are the system users adequately productive, how will the system design adapt to change, and what will happen in the case of failure.

- Advances in component technology make the choice of computer system *more difficult*, not easier. Today there are many fewer obvious constraints on what *can* be done: you can have one computer or many, and no *obvious* rule-of-thumb says which approach is right. Again, the system design must reflect the needs of the organization.

- The importance of success is growing. In the past an inadequate batch computer system was frustrating and more costly than necessary. Failure of the computer system was like having a large duplicating machine fail – it was bearable if it didn't happen too often, and you could always get a new reproduction manager and find a better duplicator. Today, a computer system is more like the telephone, or electric power – serious disruptions often directly impact the work in progress.

MANAGER'S ROLE IN THE DISTRIBUTED SYSTEMS TEAM

The EDP manager plays a key role in the team that must work together to build distributed systems:

1. **General Managers.** General management must define the direction and goals for the enterprise and set policy for information dissemination and use.

2. **EDP Managers**. The EDP manager works within the constraints and guidelines set by general management. The EDP manager factors-in the technology and technical management required to develop a system plan. The system plan reflects how the enterprise uses information, and adds in

   - the kind of computers to be used

   - the communications link between computers and users

   - the management plan for system development and maintenance

   - a plan for adapting to changing technology

The EDP manager is also responsible for the execution of the system plan and the ongoing management of information resources.

3. **System Designers and Programmers.** With a good system plan in place, system designers and programmers can focus on the many difficult problems of building distributed systems.

THE BALANCING ACT

The distributed system EDP manager uses education, judgment and experience to reach good, balanced compromises between seriously conflicting goals.

1. Service *versus* cost
2. Control *versus* usability

Service versus Cost

A cost accountant would categorize most information system costs as managed costs, meaning they do not depend on production volume, or any other obvious factor.

If information system users get everything they want, the total cost can be unbounded. Since many users don't fully understand the costs of various services, it's quite reasonable that EDP managers should be a strong influence in selecting which services should be offered.

But conversely, the EDP manager may not easily understand the full benefit of a specific service. Computer systems should be measured on the basis of cost/benefit analysis and not viewed negatively on cost alone.

Distributed systems make cost/benefit analysis all the more difficult, because much or all of the equipment may be shared among many applications, and because it's harder to find out what the total usage for specific data or services really is.

Control versus Usability

An EDP manager is ultimately measured on his ability to control operations and costs. Unfortunately, the more rigidly an information system is controlled, the less useful it is for the kind of intuitive data exploration and adaptation that extensive on-line data can provide. In other words, if you establish strict controls about adding new data items, or modifying the definition of existing data items, then you will know what data is there and what it means, but the data will be relatively useless.

What is the value of a brilliant insight due to convenient data access? What is the value of quick adaptation to change? In both cases the value may be large, but difficult to quantify. In contrast, a junior accountant can easily tally the costs of excess capacity and redundancy. Clearly an EDP manager is hired for his judgment as well as his cost-consciousness.

CRITICAL DISTRIBUTED SYSTEM DECISIONS

An EDP manager wants to establish a system plan that leads to distributed systems that are

- Functional – they do the right job.
- Effective – they do the job at a reasonable cost.
- Adaptive – they can change as the enterprise changes and as the system technology changes.

Some critical aspects of achieving these goals includes:

1. Selecting system hardware
2. Selecting communication techniques
3. Selecting vendors
4. Establishing system management style
5. Coping with new technology

Selecting System Hardware

Advances in component technology now make it possible to build information systems with system configurations anywhere between one based on a large number of low cost ($10,000) computer systems and one based on a single high cost ($10,000,000) computer. Rabid supporters of big systems or small systems will often argue that only their approach can work.

Reduced to an absurd point, the choice is between using a single large computer at a central site, linked to on-line terminals by communications lines; and using many smaller computer systems linked more directly to the on-line terminals and communicating with each other via communication links.

In fact, most good systems designers will use some collection of bigger and smaller computers, as fits the need.

The argument arises because small (mini) computers (systems costing less than $200,000) are now functionally comparable to larger (mainframe) computers (systems costing more than $200,000). In other words, the term minicomputer or mainframe says relatively little about what the system can and cannot do. Modern small computers, such as the Digital Equipment Corporation VAX-11/780 or DECSYSTEM-2020, have the kind of hardware facilities (virtual memory, extended instruction set) previously associated with only mainframes, as well as the software needed to perform large processing tasks effectively.

Although from a functional standpoint big and small computers are becoming increasingly comparable, each does something better than the other. The major advantages of using large and small computers in data processing systems are summarized below.

Advantages of big computers (costing more than $200,000). Systems using big computers benefit from economies of scale:

- *Instruction execution.* The bigger the computer, the less it costs to execute a single instruction (i.e., the processor cost divided by instruction rate is less). Hence large processor-bound applications, such as very complex scientific calculations, use the largest systems available. Most data-processing operations are not processor limited, but rather data-access limited and communications limited.

- *On-line data storage.* The lowest cost for on-line storage can be achieved by lumping many applications together on a single large system.

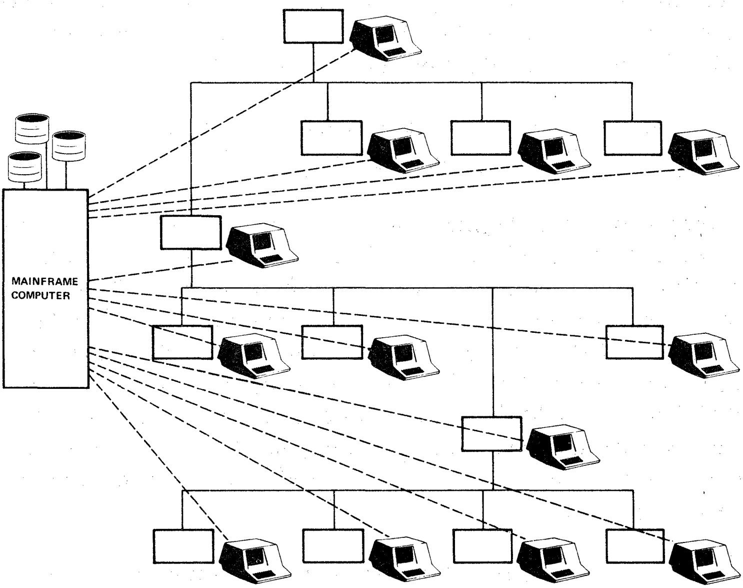


Figure 7-1. An information system can be built using a single, large computer connected to terminals by communication lines.

There are also some advantages to centralized, as opposed to physically distributed systems. Although centralized systems have been built from multiple small computers, a centralized approach usually also means using a bigger computer system.

- *System operations.* Most computer-based information systems require some support staff. This support is least expensive with a centralized facility.

- *System design.* Similarly, most systems require some number of analysis, design, and implementation specialists. These experts can be provided most economically from a central facility.

Advantages of small computers (costing less than $200,000). Although economies of scale are lost, smaller computers have their own benefits:

- *Capital costs.* If a system plan is based on multiple smaller computers, money can be invested in computer power as it is needed, in smaller increments.

- *Cost allocation.* If each major application uses an independent computer system, the cost of automating each application can be easily identified.

- *Technological advance.* If the smaller computer is part of a family of compatible products, such as the Digital Equipment Corporation PDP-11 and VAX-11 families, then as computer power is added, the incremental systems can be the most modern model, benefiting from recent advances in technology.

There are also benefits that accrue with a distributed approach – a system consisting of computers located in more than one site – rather than a centralized approach – a system with the processing hardware consolidated in one major site. Distributed systems have been built with multiple large computers. But in general, system designs that emphasize distribution of computing equipment make greater use of smaller computers.

The benefits of a distributed design includes these:

- *Communication costs.* The data and processor can be located optimally with respect to their use; thus the data communication costs are minimized.

- *Data security.* Data security problems are minimized if the data and processor are placed with the users. Access to highly secure data can be controlled by controlling access to the premises. Systems that depend on elaborate communication facilities for all data access offer more opportunities to access the data illegitimately.

- *Development cycle.* Dividing a large system into smaller, less complex subsystems leads to a shorter system development cycle.

- *Usability.* Locating the computer system near the system users minimizes the person-to-person communication problems of using the system.

- *Availability.* Building an information system with multiple computer systems minimizes the possibility that any single failure (a processor breaking) will render the total system unusable. In many distributed system designs, the key functions of a single failed computer system are assumed by other computer systems.

Selecting Communication Techniques

Communication facilities – means of communicating data from one location to another – come in a bewildering range of costs, capabilities, and styles. The subject of communications is dealt with in greater depth in Chapter 10. The problem consists of developing the communication links (e.g., stringing wires between sites or using microwave links) and developing the software that lets systems communicate with one another.

For the time being, the basic alternatives can be summarized briefly:

1. **"Roll your own."** If nothing else works, you can build communications facilities yourself. For example, some railroads have adapted the microwave links that existed along the rail lines for digital data communications. Most enterprises aren't as

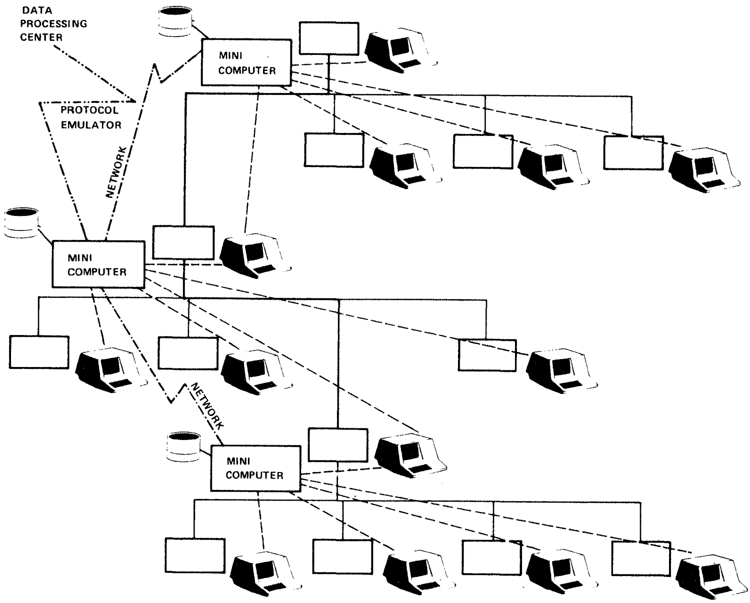


Figure 7-2. An information system can be built using many minicomputers, each serving local departmental needs. In this example, the departmental computers are interconnected into a network, and the network communicates with a central mainframe.

well equipped with existing hardware. Once the hardware exists, you still need to create the software to transmit data between systems, which may turn out to be an overwhelming task by itself.

2. **Available links, purchased software.** Several sources can provide raw communication links, in particular the phone system. Recently, some extensive communication software has become available, such as the Digital Equipment Corporation DECnet software (also discussed in Chapter 10).

3. **Specialized services.** Recognizing the complexity of building communications facilities, some companies have taken the

opportunity to set up communications services. They lease basic communication links, build a communication system using small computers, and then sell the resulting communication service. For example, they might charge a fixed amount per month for the service and charge additionally based on the amount of data transmitted. Specialized services still provide only limited coverage (you have to be where they are or important enough for them to expand to meet your needs), and the long-term survival probability of several companies is questionable.

4. **Generalized services.** The phone system is moving rapidly to develop a data network much like the existing speech network. There is no question about whether the phone system can succeed technically, but some question about what services they will be permitted to offer and at what cost (government agencies regulate their operation).

Although many different ways of communicating data are available today, the chances are relatively good that standard services will be defined and made available by multiple, competing companies. This is in striking contrast to computer hardware and software, where many vendors sell functionally comparable but totally incompatible products.

Selecting Vendors

Modern system designs make the selection of hardware, software, and communications vendors a very important choice. For example, in the case of hardware vendors, many companies can design and manufacture a computer. The sales and support for initial customers may be exceptional. But the customer ends up with the system for an extended period of time. Can the vendor continue to provide an adequate level of sales and support? Will the vendor still be making computers in another five years?

Selecting a marginal vendor (one unable to deliver fully adequate products and services) is painful enough when the computer system is used for an isolated or minor function. But if the computer is part of a distributed system, the results may be severe:

- If the computer system is used interactively, failure or malfunction may seriously affect the productivity of the system users.

- If the system provides data to other systems, symptoms of the failure may spread to otherwise reliable systems, because critical data is inaccessible or lost.

- Having system components provided by a marginal vendor can have side effects if the vendor reacts to problems by accusing other suppliers of system components ("It's the phone system!") and thereby creates additional political problems, rather than responsibly attempting to diagnose failures.

There is no simple way to evaluate the reputation of a vendor. A small vendor will often work very hard since the vendor needs to establish a reputation and business. A big vendor may be a poor provider of service, especially if computers are a secondary and perhaps unprofitable segment of his business.

You can't guarantee that a vendor will provide good service and support, but you can evaluate the probability. If the vendor claims responsive service, how will he deliver it? Where are the service people and parts supplies? How many customers does that service staff support, and how many systems? How are priorities established?

How much is the vendor investing in R&D for the kind of system you are buying? Are there people in place to fix the problems you find? When will the next version of the system be available?

Finally, find existing customers of that vendor, and ask them what the vendor is like. If the vendor looks golden, but you can't find contented customers, there has to be a problem.

Establishing a Management Style

Managing a large, distributed information system is a challenge, much like the challenge encountered when a company builds its first remote plants. With many different applications being developed and maintained, and computer equipment in many sites, a system management style can be established anywhere between completely centralized control – with all decisions made or closely reviewed centrally – (we'll call that "autocracy") and completely decentralized control – with decisions made primarily by the different system user groups – ("autonomy"). Carried to an extreme, neither style is ideal.

An autocratically-managed system will probably end up being highly reliable, with well-controlled costs, but may be of little use. An autonomously-managed system will be very useful locally (to each department) but there will be little or no system for the whole enterprise. The most effective management style for most systems lies between these two extremes.

System Management Style Isn't Dictated by Hardware Choices. Sellers of large systems like to say that using smaller, remotely located computers is bound to lead to anarchy. But many examples exist that belie such claims.

For example, a large insurance company uses many small distributed computers as functional tools. One system supports Workman's Compensation Claim Processing and nothing else. That system has no programmers and no separate management. The system was designed by an analysis and programming team working with the department for a developmental period and then moving to other tasks. The day-to-day operations are managed by the supervisor of the department.

Computer network technology makes it possible to operate a distributed computer system as if it were a single machine. The communications links between the machines can be used to specify which programs are run on remote computers, as well as to transmit data between programs. All programs could be developed and maintained from a central site and the remote systems operated centrally as well. This approach has many of the benefits of totally distributed systems – ones in which the remote systems are controlled remotely – but maintains centralized control of the entire system.

Conversely, the use of a large central computer doesn't mean that the operation will be well managed, nor that it has to be managed in a centralized manner. The system manager can act as a wholesaler, and sell bulk computer resources to individual departments, each of which build their independent applications on the single system.

System Management Style Should Match Enterprise Management Style. Although a wide variety of system management styles are possible, in most cases the computer systems should be managed in a style that reflects the overall management style of the enterprise. If the enterprise is centrally managed, with few decisions ever made without central review and approval, then it would be questionable to seriously consider a highly autonomous computer system management style.

Similarly, if the management style of the enterprise encourages distributed decision making it would be questionable to try to establish a highly autocratic computer system management style.

Specific Management Decisions. The specific decision areas that affect a distributed system are listed below. They are ordered such that those with the greatest impact on total system operation are first. A system manager believing in the autonomy of the remote systems would focus overall system management on a few areas near the top of the list, and leave other choices to local department managers. In a more autocratic management style, additional areas would be managed centrally.

1. *Application design style.* If the system is going to work as a whole then a unifying set of concepts must be developed. For example, since the various applications constituting the distributed system will share data, there must be some agreement on how data is shared.

2. *System interface style.* If individuals are to use more than one application, or individuals move between departments, it will be very beneficial to have some commonality in how the different applications interact with the user. At the very least, the knowledge of one application should be a help, not a hindrance, in the use of another application.

3. *Computers and operating systems.* The greater the variations in computers and operating systems, the harder it will be to make the system effective as a whole. Nevertheless, a strong concept of application design and system interface style can make the choice of specific computers and operating systems less important. Conversely, using only one computer and operating system could have little value if no attention is given to the questions of application style.

4. *Data base.* The data in the system is the ultimate resource. But without any control or management, one system's data may be another system's gibberish. Strict control over all data (e.g., requiring all data definitions to be granted centrally and not changed without approval) makes the data highly usable across the systems, but makes the data too cumbersome to use effectively for many informal applications. Having no control over data makes it optimally useful for informal, local applications but of no general value.

5. *Programming languages.* Minimizing the number of programming languages used makes programmers more productive. If applications are written in a single, standard programming language, then there is the greatest flexibility in assigning which programmer is to work on a specific application. Minimizing the number of different programming languages used also maximizes the ease with which a program can be moved from one computer system to another.

6. *Specific application design.* Reviewing each application design centrally is one way of assuring a uniform style in structure and interfaces and enforcing central control on the data base design.

7. *Specific application programming.* The strictest control possible is to supervise the coding of each specific application.

Coping with New Technology

Computer technology has advanced at a bewildering rate in the last two decades. Even computer vendors have had trouble adequately planning for what the future would bring. Although advances have been rapid, in retrospect the rate has been remarkably constant. Chapter 11 discusses trends in technology in greater detail.

But being able to predict the rate at which cost/performance will improve is valueless unless the improvements can be related to the productive impact on the enterprise. Conversely, if the economics of computer system usage are well understood, technological progress is relatively easy to deal with. The next section gives a model for understanding computer system productivity.

COMPUTER SYSTEM PRODUCTIVITY

Computer systems should be evaluated as productive tools. Money should be invested in a computer system (as opposed to a drill press or new machinist) only if the computer system is the best investment (has the highest return). The money saved by a computer system is computed as

$$\text{return} = \text{costs displaced} + \text{new revenue} - \text{new costs}$$

The costs displaced are the easiest part to analyze. For a cost to be displaced it must disappear when the computer is added. Examples would be the cost of maintaining an old system that is removed, or the costs of clerical staff that is no longer needed with the new system (but you can't displace a fraction of a person – for labor costs to be displaced you must end up with fewer employees). Often it is future costs that are displaced, in the sense that the computer system increases the productivity of the existing staff; thus the workload grows without a need for *additional* staff.

New revenue can be predicted, and the prediction measured over time. Except where the system is a product, such as when a bank adds automatic banking machines, new revenue will probably not be a dominating factor in the equation.

New costs are often underestimated before the fact and are the area in which technological advances have had the greatest impact. For example, the cost in individuals using computers is generally undervalued. This mistake becomes more critical as the cost of the hardware goes down.

Total System Costs

Not only are you interested in the total costs, both direct and indirect; you are also interested in the costs over the lifetime of the system. If these costs can be understood, then the impact of technology can be seen as three major thrusts:

1. Hardware is becoming less expensive. The cost to provide a certain computer function will continue to go down over time.

2. Labor costs are going up. The value of the people writing software for the system, operating the system, and using the system will go up, particularly when compared to the value of the hardware.

3. More people will be affected. As the hardware becomes less expensive, it will be used more extensively. Greater usage tends to act as a further multiplier on the value of the people using the system.

The net conclusion is quite simple: *it is increasingly important that the system does the right job and leads to the productivity of the user, and decreasingly important that the usage of the hardware is optimal.*

Total Life-Cycle Cost Model

Most of the key system costs are identified here. Once they have been analyzed in the context of a particular application, they can be estimated over time by understanding their dependence on hardware technology and labor costs and their impact on user productivity.

1. **Site preparation.** The equipment must be located somewhere. The costs may include special environmental considerations such as adding raised floors, physical security, fire protection, and wiring. The computer may require additional power and air-conditioning.

2. **Hiring and training operators.** The computer may need an operational staff. This may be a new task assigned to present employees or new individuals may be hired. In either case, special training will probably be needed.

3. **Hardware costs.** The equipment must be paid for.

4. **Hiring and training programmers.** A new system may require additional programmers or special training.

5. **System development.** An application may require special hardware or custom system software.

6. **Application development.** Except for the few cases in which a totally "turn-key" system is used, some specific application development (programming) will be required. With the low cost of computers, *it is very easy for application programming to cost more than the hardware.*

7. **System maintenance.** Over the life of the system a substantial investment must be made in preventive and remedial maintenance. The total maintenance investment may equal the initial cost of the system.

8. **Program maintenance.** Although software doesn't break, there are always some design and implementation errors to be corrected. Over the lifetime of the application, there will be many desired extensions and adaptations to the initial software. *The total software maintenance cost may be many times the initial cost to develop the software!*

9. **User training.** Although a system may be billed as "easy to use," most systems will either require formal training for users or result in a loss of productivity when initially used.

10. **User effectiveness.** As was noted above, for many present and future systems the real value is in the productivity of the on-line users. A small change in user productivity may have a large impact on total system costs. Chapter 8 discusses the question of system usability at length.

11. **System failure.** All systems occasionally stop delivering useful service. Computer system failure may have little or no impact on the system users, or it may cause some loss in user productivity, or the users may become nonproductive. Computer systems can be made more reliable by a greater investment in hardware and software. As hardware costs decrease, and the impact of failure increases because more individuals depend on system operation, it becomes worthwhile to design systems that have maximum reliability.

12. **Evolution.** Given the rate of technology evolution, any system acquired in the near future is almost sure to become obsolete. Part of the cost of a new system will represent the retraining of operators and users, and the conversion of programs. Evolution costs can be minimized by anticipating them from the beginning.

Minimizing Life-Cycle Costs

Effective EDP management is aimed at delivering high-value system service while minimizing total system costs. Many of the distributed system design issues (Chapter 12) and design techniques (Chapter 13) are the direct result of the life-cycle view. For example, system responsiveness and data integrity – the two most difficult and pervasive design problems – both contribute to system cost reduction. Responsiveness is key to user productivity; data integrity is necessary for the system to be useful and to minimize failure costs.

A total-cost approach also leads to the following conclusions:

1. Plan for the long haul.
2. Build on existing systems.

Plan for the long haul. Historically, computers have represented large capital acquisitions. This meant that most enterprises carefully considered the initial purchase, but all too often forgot about the ongoing costs. As the cost of equipment diminishes, it becomes more and more important to focus on the day-to-day costs and benefits, rather than on the initial acquisition.

Although applications evolve over time, specific pieces of hardware and individual programs tend to stay around for much longer than was initially planned. The initial design of a program is often provisional – to be improved over time. But over time new opportunities arise that take priority over fixing a program that works, and the initial provisional design lasts and lasts. The moral is to view actions as permanent, unless the time and effort to change them are firmly committed.

Build on existing systems. Existing systems represent a very valuable resource. When a new application is needed, every effort should be made to utilize the investment in existing systems. Conversely, when a new system is designed, the possibility that it will serve unanticipated needs in the future should be seriously considered.

SUMMARY

The EDP manager is a key member of the distributed systems team. As the use of an automated information system increases, its importance to an enterprise grows and along with it grows the importance of good EDP management.

EDP management means balancing conflicting needs. Extreme approaches are unlikely to succeed.

Focussing on the total life-cycle costs of a system puts many of the costs and cost trends into a valuable perspective. In particular, using a total-cost approach makes it possible to deal rationally with rapidly advancing computer technology.

Chapter 8
USABLE SYSTEMS

The usability of a system is a measure of the productivity of the system user[1]. It is not an absolute measure, but there can be wide differences in usability between systems that perform similar functions. Usability is important in all systems, but the user working interactively is perhaps more aware of usability than is the batch user, so our discussion assumes that the user is at an on-line terminal, directly connected to the system.

WHY IS USABILITY SO IMPORTANT?

The costs of a computer system consist of the costs of the hardware, the costs of application software development, *plus* the costs of using the system. Historically, the costs of the hardware dominated, and much effort was made to maximize the productivity of hardware use. But continuing improvements in computer component technology have changed the situation so that now the productivity of system users has a major and growing impact on the cost-effectiveness of a computer system. The reduction of hardware costs as a result of improvements in computer technology has directly led to changes in typical computer usage.

---

[1] The consideration of usability must run throughout the design and development of any computer system. The usability of a system to the programmer and operator is also important. Since this handbook is written primarily for computer end-users, not those who build tools for programmers, the emphasis is on user productivity, not programmer productivity.

1. Wider use of computers: more people use computers, and a diminishing percentage of them are computer specialists.

2. More interactive computer usage: as hardware costs diminish, more computers are used directly, via a terminal.

3. Computers do more of the job.

Under these circumstances, the value of usability is probably increasing more rapidly than the cost of computer hardware is decreasing.

HOW TO TEST USABILITY

The less you know about computer systems, the better you can determine the usability of a given system. That is because someone who knows a lot about how computer systems work may automatically compensate for a system's lack of usability.

The way you test a system is by using it. Systems will fall into four general categories of usability:

1. **Immediately usable systems.** With some systems, an individual may become productive in a very short period of time. An example is an automatic banking machine that produces cash when a valid bank-supplied card is used, and a correct transaction entered. This kind of system does not depend on prior computer experience.

2. **Usable with self-training.** Another class of systems requires some amount of preparation, such as reading a manual. An example is the modern word-processing systems available today, which can be used by an experienced typist without any formal training beyond reading a manual.

3. **Usable with training in new techniques.** The next class requires some amount of training beyond reading documentation. The additional training may come from formal courses, or informally from other system users. An example of this kind of

system is a fancy office copier, which can't quite be mastered by just reading the usage guide, especially when it comes to special conditions like paper jams.

4. **Usable with application training.** Finally, there is a class of system that requires a fairly detailed understanding of the entire application domain. For example, most large computerized typesetting systems are designed within the unique jargon of the printer and will only be usable when a large part of the entire printing process is well understood.

In general, the less training required to use a system, the better. In some cases training is required because of the job being done, which is a different matter. But if two systems do the same job, one requiring less training is clearly less costly, all other things being equal.

CHARACTERISTICS OF USABILITY

Usability is a property of the system as a whole. Some key factors contribute to a system's usability.

- Functionality
- Simplicity
- Predictability
- Responsiveness

Functionality

To be usable a system must perform the right functions. You can provide application functions in various ways, some more usable than others. For example, if what the user wants to do is logically a single function, it is much more usable to provide a single computer system function than to provide a set of more primitive functions that must be used in a particular sequence to achieve the desired end. That may seem obvious, but programmers are used to finding circuitous ways of getting programming tasks done and may not be adequately sensitive to the needs of a non-expert user.

Simplicity

Simplicity is another way of looking at the same problem. If there are many ways of providing a particular application function, the one that is simplest *to the system user* is almost always the best.

Predictability

If you have used a system, but not a particular function, a good test of its predictability (for you) is to try to guess how to do the new function. If your guess is right, or close, then for you the system is predictable. The ideal predictable design is one that can be used with no knowledge of the system beforehand. An example of designing a system to be predictable is making system commands default to the most common case. In other words, if you present a command with as few optional inputs as possible, does it do the right thing the majority of times?

Responsiveness

The characteristics discussed above concern how easy it is to do a particular function. Responsiveness is a measure of both how rapidly a function is performed, and how predictable the performance is. A system with poor responsiveness either delays the user, which obviously detracts from productivity, or makes the user wait for a response which is sometimes very quick, but sometimes slow, in which case the uncertainty detracts from productivity.

The response delay time (time from making a request to the system until the response is received) should be both adequate to the task and related to the complexity of the request. A delay that inescapably delays the system user is detrimental to productivity. For example, in a supermarket checkout system, any system-imposed delay that keeps the checker from processing groceries has a direct penalty on system productivity. In other types of systems, a delay may be acceptable because the user can do something else while the request is being processed.

The response delay should correspond to the complexity of the request, from the user's point of view. It is very disquieting if the response to a simple request takes as long as the response to a complex request.

Finally, responsiveness should be predictable, to the degree possible. In almost all interactive systems there will occasionally be long delays, just as occasionally the telephone system cannot complete your calls. What counts is the average length of delay and how often requests go beyond the average. In most cases, it is better to have a slightly longer average, with fewer requests extending substantially beyond the average. Users will grow accustomed to the average but become impatient during exceptionally long response delays.

Responsiveness is often measured as the time at which 95% of all requests are complete (the mean is the 50% completion time), since the 95% measure captures both average response and a feeling for the number that deviate beyond the average. A totally predictable response would have an average, a mean, and a 95% response measure all equal to one another.

ACHIEVING USABILITY

Many approaches to achieving system usability have been implied above, such as making system use simple and predictable. Some additional techniques are discussed below.

Interaction

Our discussion of usability has been based mainly on interactive systems. Let us here briefly summarize some of the major benefits of interaction.

What do we mean by "interactive"? As we use the term here, "interacting" with a system is comparable to holding a conversation with another individual.

It could be argued that any system is interactive, because the user "interacts" with the computer. We use the term here only for systems used in a conversational manner. Interactive systems are typically designed with an on-line terminal with which the user interacts with the computer systems.

Interactive system design has many advantages:

1. Speed of computer response.
2. Immediate identification of many errors.
3. Minimum knowledge of internal details.

Speed of computer response. Interactive systems can get the job done faster. For some applications this is a critical factor. Being able to print payroll checks on demand has little value in most businesses, but being able to check credit in a few seconds is critical to credit authorization systems.

Immediate identification of many errors. The application program is usually written to detect errors in the format and content of user inputs, and return that information. There are two major benefits from learning about mistakes quickly:

- The system is "friendlier" to use, because less time is lost when an error is made; the user is less concerned that each entry be precisely correct.

- The data entered tends to become more accurate, since obvious errors, such as inconsistency with previously entered data, can be detected and corrected while correction is easiest. For example, if the terminal user is entering data from a phone conversation, and the data system warns that input is inconsistent, the user can ask further questions over the phone to clarify the problem.

Minimum knowledge of details. The user of a well-designed interactive system need not know all the details, because the system can help. For example, the system can display the commands that are acceptable at any particular point and instruct the user how to choose among them.

Problem-Oriented Design

A basic approach to making computer systems usable is to hide as much of the internal operations as possible. For example, the telephone system is one of the most complex computer systems imaginable, in which a remarkable number of functions can be directly commanded from the user's terminal (the telephone). But no one thinks of the telephone as a computer, just as a "simple" communication tool.

Applications are more usable if they seem like part of the environment they serve. This means designing with a careful attention to the natural work flow, choosing the correct terminology, and concealing as many extraneous computer details as possible.

Menu-Oriented Design

The concept of having the system show the user what can be done at a given point is called "menu selection" (a restaurant menu is a prepared list from which you select what you want). Menu selection systems are particularly well suited to applications where the choices can be pre-set. For example, a menu-oriented design would work well for entering drug orders at a pharmacy, since the set of drugs is limited, as are the dosages and means of giving the drugs.

The menu-oriented design is valuable in many ways:

1. The user sees what can be done, rather than having to memorize the choices.

2. The correct sequence can also be seen from example, rather than remembered. There is no more question as to whether two options must be separated by a blank or by a comma.

3. Further rules can be enforced transparently. In the drug example given above, invalid combinations of drug and dosage are reported as errors.

4. If desired, the interaction can be done by literally pointing at the selection items (with a lightpen or touchscreen and a CRT

terminal display). This technique eliminates the need to type, which has value where typing cannot be assumed as a skill, or where having to sit down and type would substantially slow the interaction.

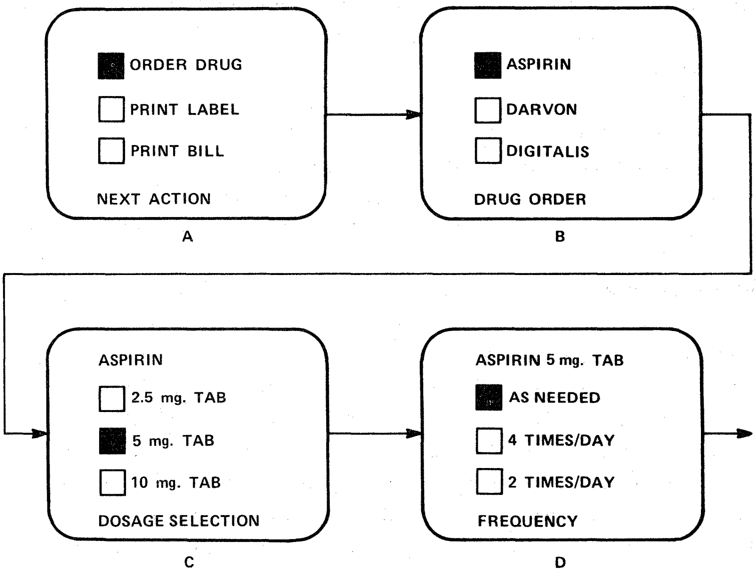


Figure 8-1. A computer system for a pharmacy is designed as a collection of menus from which the user selects an action: (a) The user wants to order a drug; (b) ASPIRIN is selected; (c) 5 milligram tablets are selected as the dosage; (d) The drug is to be administered AS NEEDED.

Forms-Based Interaction

For application functions that have too much text entry to be constructed as pure menu choices, the interaction can be structured as fill-in-the-blanks forms, which are a cross between pure text schemes and menu schemes.

The form, which is typically displayed on a CRT terminal, performs many of the functions of a paper form:

1. The terminal user sees what items must be specified, and is given some hints as to the correct form.

2. The input is received by the program in a standard, easy to interpret form.

Additionally, a computer-based form provides benefits that are not available with paper forms:

1. The data entries can be validated for syntactic correctness: Does the ZIP code have letters in it? Is the account number check-digit correct? This processing is simple enough to be provided by an intelligent terminal in many cases.

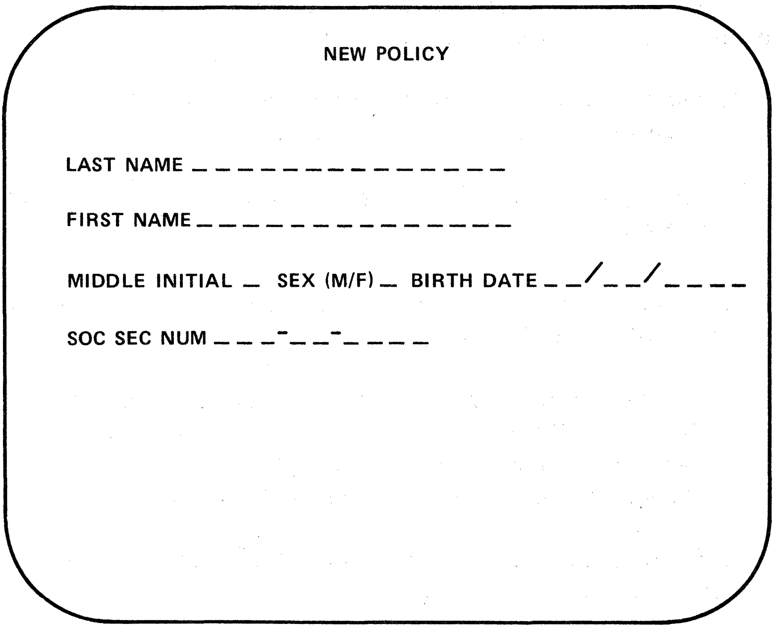
2. The data entries can also be compared to the on-line data base and checked for consistency.

Both of these error checks help to increase the accuracy of the data, once it is stored in the on-line data base.

"Natural" Language Interaction

A long-standing computer industry goal has been a system that you could talk to, or at least write to, in a natural language, such as English. An English-language interface would be simple and predictable for English-speaking users, and thus very usable. Highly usable systems use obvious terminology, which is natural language except where the application area uses highly-technical jargon, and often has language-like command structure, with "verbs" and "nouns."

But the goal of being able to use a natural language as a means of conversing with a computer, which is quite different from using languages that *resemble* natural languages, has eluded all so far. At the root of the problem is the tremendous complexity of natural


NEW POLICY

LAST NAME _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

FIRST NAME _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

MIDDLE INITIAL _ SEX (M/F) _ BIRTH DATE _ _/_ _/_ _ _ _

SOC SEC NUM _ _ _-_ _-_ _ _ _

Figure 8-2. In a form-based system, the user completes forms on a terminal. The forms appear very much like paper forms. The system provides immediate checking to be sure the data fields on the form are reasonable.

languages. To understand what sentences mean you apply knowledge that goes beyond word definitions and grammar. For example, consider these two sentences:

1. "Time flies like an arrow."
2. "Fruit flies like a banana."

Is it reasonable to expect that a machine would understand how different these sentences are?

Nonetheless many systems are sold as English-like, and if you watch an expert using the system, the communication language might look like English. Beware – it's like the horse that can count by following subtle hints from the trainer, and pawing the ground a certain number of times. To be more specific, there are two very different cases to consider:

1. In some systems the valid input commands are also proper English sentences, or very close. This is a good attribute, since it makes commands easier to remember, and makes the result of a computer session more self-documenting (you don't have to be an expert to understand what happened). But,

2. No existing system comes close to recognizing arbitrary English Language input. There are hundreds of grammatically correct ways of phrasing any request – no existing system will recognize more than a few of these.

In other words, the users of "natural" language systems have to learn the details of the computer language just like anyone else. These systems will not accept input from untrained individuals, except by chance. There are advantages to this kind of design, but the systems don't perform magic. They don't understand language.

Application Documentation

It's easier to design and build a good information system than it is to create the documentation and training materials for system users. Most of the application programming is never seen by anyone but the programmer. As long as the logic is correct, it makes little difference whether the program construction is elegant. In contrast, all of a user manual is seen, and every poor sentence will detract from its value.

Just as programmers are poor evaluators of program usability, they are poor documenters (there are exceptions):

1. The programmer is tempted to explain how clever the application is internally, whether or not this has any value to the user.
2. Conversely, many of the subtleties of application usage are totally obvious to the programmer who has created it but obscure to other people.

Application programmers should be hired for their ability. If one can also write well all the better, but it isn't reasonable to expect a good programmer to be a good writer. Critical documentation should be produced under the direction of a professional technical writer.

Standards and Invention

An individual in an enterprise will use many different computer applications. Many individuals will use more than one application in their day-to-day work; over time most will use more than a single application. The mutual consistency of these applications will affect the user's productivity. The consistency between applications may even be more important than the usability of specific applications.

Consistency means compromise to the programmer. If a new design is to be consistent with an existing design, then it may have to replicate some of the known shortcomings. Accepting compromise can be very difficult for a programmer who is trying to design the best possible program. But consistency can only be achieved by identifying critical aspects of applications, establishing standards for their design, and then managing application development to those standards.

Balancing invention against standardization is a delicate art, requiring substantial judgment and maturity. Without progress much of the benefit of advanced technology is of little value. With unconstrained progress, every application is different and nothing works well together.

SUMMARY

The quality of usability depends on how much training the user needs, how much help the system can give the user, the speed of system response, the simplicity of the system's language, and many other factors, some of which are invisible to the user.

Chapter 9
INFORMATION WORK STATIONS, A DESIGN MODEL

The design of any computer system has three general goals:

1. The application meets existing functional needs.
2. The application is easy to adapt to future, changing needs.
3. The design model is comfortable for the analysts and programmers doing the system engineering.

This chapter presents a high-level model for building distributed systems. This is a general discussion, not a cookbook solution. Some critical problems identified here can be solved only by the system designer for the specific application.

The model gives a conceptual framework for describing distributed systems. Use of any model provides a common framework for all individuals participating in the design and development of the system, thus facilitating communication. This particular model helps focus attention on critical problems.

The model has two key concepts: the work station as a way of viewing the informational aspects of organizations, and the *information transaction* as a way of viewing information flow. The model is applicable to most applications:

- Information work stations are derived from common industrial engineering practices; a similar concept has been used to analyze manual clerical activity.

- Much of the existing information flow in enterprises is already in transactions.

The model has no direct implication on the number or size of computers in the distributed system or their location. Its function is to describe the information flow and processing in the application. A good understanding of the information flow makes the selection of computer equipment much easier.

SIMPLIFYING THE PROBLEM

The most important step in designing a distributed system is decomposing the entire system into a number of smaller pieces – information work stations. Manual information processing systems typically have been broken into work stations – clerical departments. Each work station focusses on a particular task or set of tasks and is able to do that task effectively. The individuals working within a given work station cooperate effectively, since they share a common view of the work. Between work stations, less effective, formal communication is used.

The choice of work stations is important. If there are too few work stations, each will be large and handle many different tasks; such a concentration detracts from effectiveness. If there are too many, common tasks will be split into separated work stations; this dispersion also reduces effectiveness.

Computerizing information systems doesn't reduce the need for problem simplification. A computer can store and process large volumes of data. But the definition and use of the data must be carefully managed, with strict procedures and controls, in order to avoid the data base becoming unreliable and inconsistent. In turn, this strict management makes it much more difficult and time consuming to change or adapt the data base. Thus the strict management needed to guard the value represented by the data base also makes the data less useful.

A smaller data base can be managed with less formal controls. The data can be kept consistent and reliable, yet can effectively serve the needs of the system users.

The work stations are major boundaries within the total information system. A computerized work station is a small, local information system. The user of the computer sees primarily the programs and data of the work station. Each computer application can be tailored to the usage style of the local work station. There is no reason why the payroll work station, and the Treasurer's work station must be similar, as long as they can interchange data when needed.

INFORMATION WORK STATIONS

Work stations are the information system equivalent of the departments in an organization. Departments are formed so that the organization can be managed; work stations are formed so that information flow can be managed. For clerical activity, work stations and departments are often the same.

In the past, all work stations were manual (not automated) and work stations worked together by interchanging paper forms, by phone conversations, etc. In the future, more information systems will include automated work stations that use computers and communicate data electronically – the focus of the model.

A work station processes a specific type of information. A clerical department (a manual work station) is responsible for some collection of transactions. The choice of which transactions are processed by a particular department is based on

- what data is needed to process the transaction (the data base)
- which transactions are logically related

An example is the accounting department of an enterprise.

The specification of work stations is influenced by the size and complexity of an organization, and the intent of the information system. In a very small enterprise, the accounting department is part of a single department called the "front office". In a larger enterprise, the accounting department becomes many departments, such as the accounts receivable department. In a very large organization, the accounts receivable function may be broken into departments that deal with customers on a product line or geographical basis.

If we focus on the informational aspects only, a work station is characterized as a function that

- is viewed as a single maildrop
- performs a well-specified function for the rest of the enterprise
- receives messages at the maildrop from other work stations in the enterprise
- as a result of receiving a message, performs specific information processing by using defined procedures and a local data base, and optionally by sending and receiving further messages.

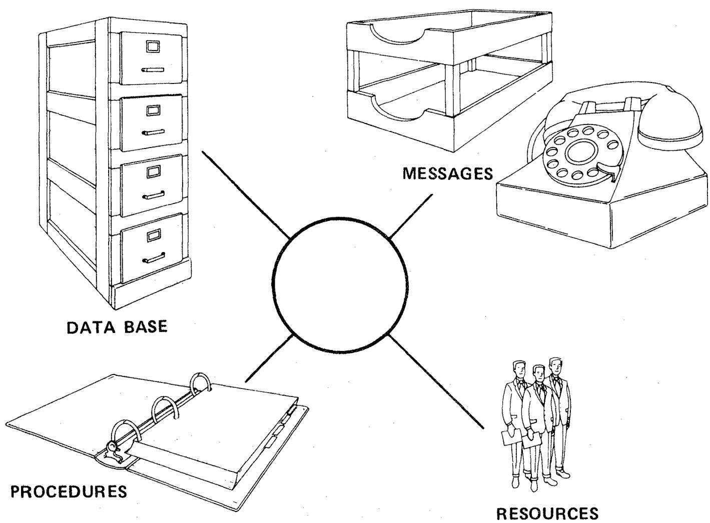


Figure 9-1. A manual work station receives paper and telephone messages. The messages are processed by using local resources, a local data base, and a set of well-defined procedures.

In designing a distributed application we first try to understand the informational needs in terms of work stations, and then automate some subset of these functions by using computers and data communication equipment.

Any enterprise can be described in terms of work stations. Normally the information structure, expressed as work stations, closely matches the organization of the enterprise. For enterprises with formal management and well defined procedures the work station definition may already exist.

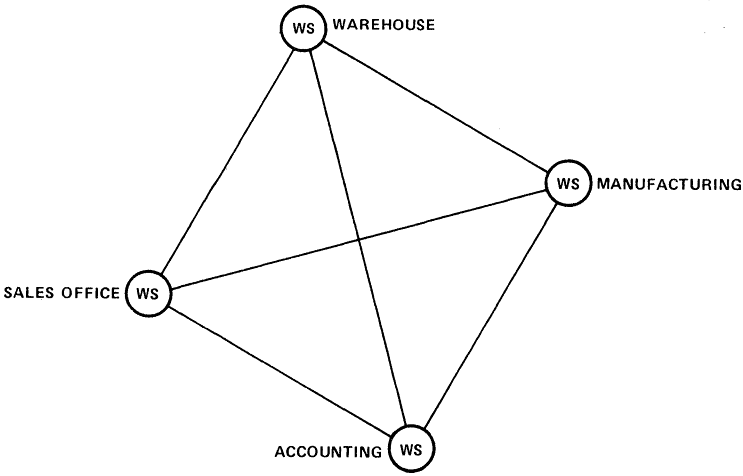


Figure 9-2. The clerical departments of an enterprise form a network of work stations which interchange messages to perform cooperative processing.

For small or informal organizations some assumptions may have to be made as to what constitutes a work station, or whether one person doing a number of independent functions should be viewed as a single or multiple work station. Analyzing any activity helps to manage it and to uncover inefficient or obsolete means of information processing.

BUILDING AUTOMATED APPLICATIONS WITH WORK STATIONS

Once the informational needs of an enterprise are well understood in a work station model, the areas that can use computer processing to advantage become evident.

The first step is to add a computer within a department. The department continues to appear unchanged externally; internally a computer performs some of the information processing tasks. Typically, the accounting department is one of the first to install computer processing. Other departments may still send messages via mail to accounting, but when received, the messages are keyed into a computer, where an accounting program processes the data and generates responses.

Usually, the second stage in automating information flow is to provide direct terminal access to the information system of another department. For example, when a sales invoice is generated by the computer system that supports the sales department, the clerk preparing the order can also initiate a transaction in the accounts receivable computer system. The different systems may have separate terminals, or a single terminal may be used selectively for different applications.

A third stage is to interconnect the departmental systems, so some messages go between systems, as well as between users and systems. From this point on, computer power can be added until the information flow of the entire enterprise is automated, and perhaps the enterprise is electronically coupled to the outside world.

The Automated Work Station

The automated work station was informally defined above. In a more precise sense, an automated work station is *an information processing system* with the following characteristics:

- **Well-Specified Function** – Like its manual counterpart, the automated work station performs a well-specified function, as viewed from the outside, such as accounts receivable.

- **A Set of Standard Procedures** - Like its manual counterpart, the work station follows defined procedures. For the manual work station there may be a written procedure book. For the automated work station there are computer programs.

- **Internal Data Base** - Most work stations maintain a private data base that is used locally. In general, (like its manual counterpart) the internal data base of a work station will be accessible externally only by sending a message to the work station and getting a report (see more below).

- **Communicates with the "Outside" via Messages** - Work requests are received either manually (mail, memos, paper forms), or via electronic communications.

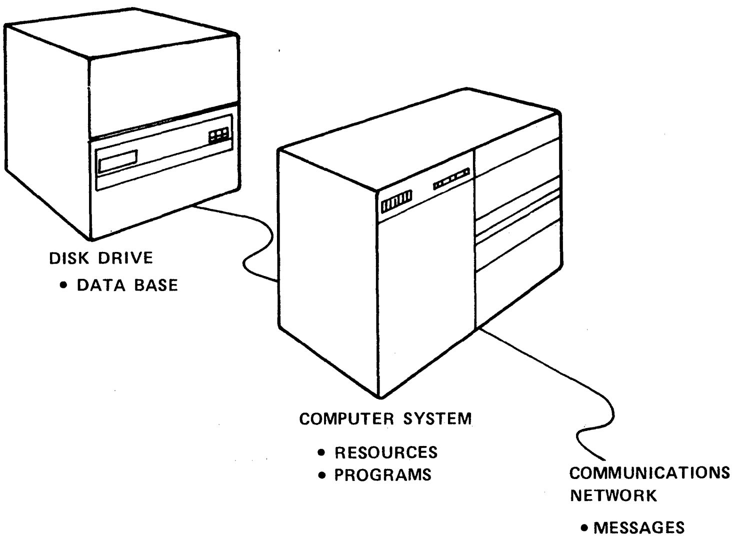


Figure 9-3. A computerized work station has the same components as a manual work station. Messages are recevied by electronic data communications. The resources are the processing and data storage subsystems. The local data base is on-line data storage. The well-defined procedures are computer programs.

Computers and Work Stations

There is no simple relationship between a work station and a computer.

Work Stations That Use Part of a Computer. Multiple work station applications may share a single computer by using conventional multiprogramming techniques. Whether such work station applications use internal (conversational) or external ("post-office") communications to interchange data, *each application is designed as a separate logical function.* The definition of work stations partitions that application logic, and suggests how computer hardware should be used.

Work Stations That Use a Dedicated Computer. A work station may have its own unshared computer. With a dedicated computer there are no questions of how computer resources should be allocated, whether the computer can be moved, etc.

Work Stations That Utilize Multiple Computers. Finally, a work station may use multiple computer systems.

- to achieve performance that is unavailable with a single processor
- to provide a highly available work station function, with one system available to substitute for another in the case of failure
- to achieve effective operation in an application that is viewed as a single function externally but is actually implemented in many separate operations.

SPECIFYING WORK STATIONS

The work station concept helps to organize information flow, which is the essential problem in automating information systems.

Decomposing an information system into work stations is like organizing an enterprise; any specific organization will have advantages and liabilities. In either case you must understand what you want to accomplish.

Computerizing work stations increases the importance that the work stations accurately reflect the information processing needs of the enterprise. When the work station is automated, the work station design affects the ease with which work can be done. Once an investment is made in computer equipment and programming, it becomes harder to change the definitions.

Many factors are involved in work station specification:

Division of Labor

Dividing an information system into work stations helps clarify information needs, costs, and responsibilities. It becomes clearer which applications have the greatest benefit to the enterprise.

Information Control

Work station specification and aggregation (placing several work stations under common management) helps clarify information management assignments.

Anticipated Growth

Farsighted choice of work stations greatly eases the costs of later organization growth. If work station functions are clearly delineated, the information systems can grow with the organization.

Anticipated Reorganization

In the same sense that a well designed organization anticipates, to some degree, the changes that will be necessary with growth, an information system can be designed that will adapt gracefully to those changes. The boundaries between work stations reflect the probable future organizational divisions.

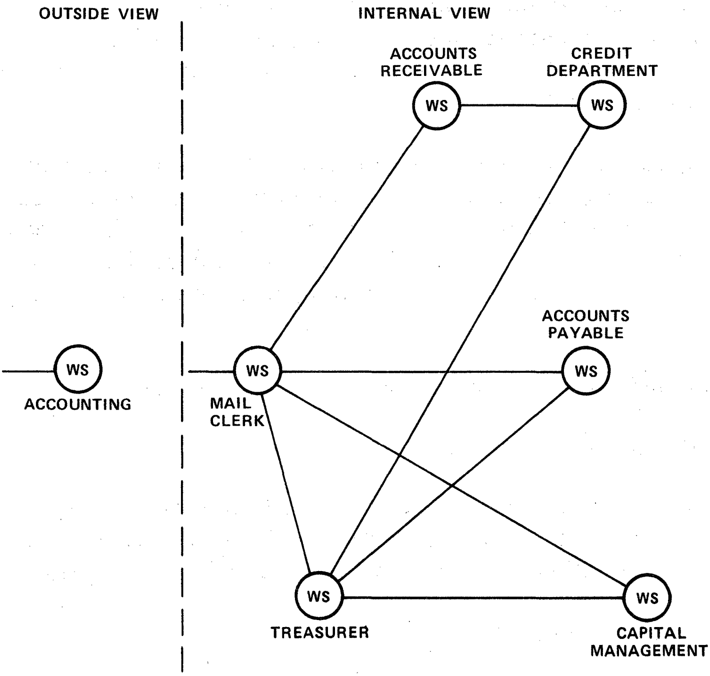


Figure 9-4. A well-specified work station lessens the data processing problems due to organizational change. In this example, the accounting department has grown substantially since the system was designed, but messages can still be sent to the accounting work stations. An internal mail-clerk work station examines the messages and dispatches them to internal work stations.

Managed Information Costs

Some information costs are *managed* in the sense that they are not directly related to the activity of an enterprise. For example, a business could install an econometrics information system to be used in strategic planning. The amount of econometrics calculation has no obvious relationship to the amount of product produced nor is it obviously bounded, like the lighting costs for a building. One way of controlling such information costs is to identify a specific work station to provide such service, and limit the computer resources available to that work station. If load increases to the point where resources are exhausted, the level of service will degrade. At this point management can choose to add more resources, or to allow the degraded service to act as a control on usage.

MANAGEMENT INFORMATION SYSTEMS

Automating an information system has the additional benefit of providing a powerful tool for managers – a management information system – by providing rapid and flexible access to the operational data. Work stations provide a natural organization for a Management Information System (MIS). In an effective management organization, each level of management provides intelligent information processing by accumulating status information from the local department, and from subordinate departments, abstracting key themes and summaries, and passing these upward in the organization.

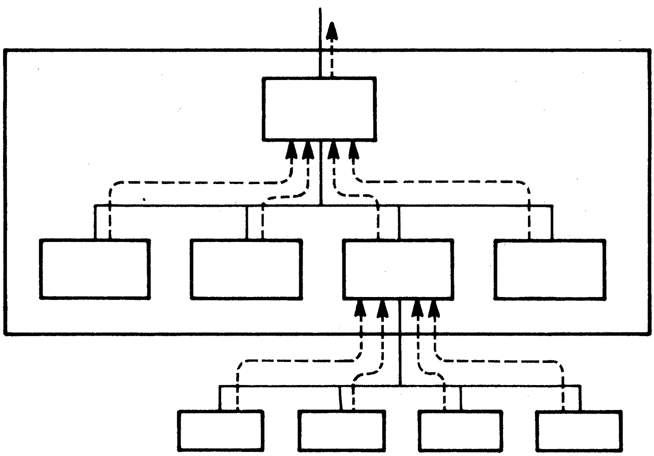


Figure 9-5. A work station system design accurately models the process by which management reporting information is abstracted and filtered at each level in an organization. A work station orientation is well suited to the design of a management information system as well as a conventional data processing operation.

Similarly, as management direction comes down through the organization, each level of management expands the direction into more complete operational detail.

A management information system that is structured into departmental work stations naturally reflects the information processing that must occur in an effective management information system. Some early, unsuccessful attempts at building a MIS failed because the system provided only operational data, without middle management abstraction. It does little good for the Chief Executive Officer to know how many boxes of which product have been loaded on specific trucks.

INFORMATION TRANSACTIONS

The other major concept in the model is the information transaction. Within a complex information system, information should be interchanged in well-defined transactions – concise and complete conversations.

Transactions are not a data processing invention. Most functional information interchange is in transactions: making a purchase, paying a bill, placing an order, requesting information, etc.

Information transactions are particularly valuable for communication between individuals doing separate but interdependent jobs – the communicatons can be packaged into well-defined conversations that minimally disrupt the work.

Information transactions have the following characteristics:

- a well-defined purpose
- a well-defined beginning and end
- usually, a clear success or failure

A specific sales order is an example of a transaction. The purpose is well defined, and has a beginning and an end. The sales order succeeds if the customer is known to have approved credit, and if pieces desired are available.

Breaking information interchange into transactions is very important in computerized information systems.

- Decomposing complex processing into specific transactions makes the system more understandable.

- Structuring processing into transactions helps build reliable systems.

The identification of work stations partitions the data and programs that constitute the system into more manageable subsystems. The identification of transactions divides the data processing into small complete pieces. These simplifications make each piece easier to understand, easier to automate, and easier to maintain over time.

Structuring processing into transactions improves data reliability by identifying points at which the data base is consistent. A single sales order modifies many data records: the customer's account is debited, inventory is diminished, a receivable account is created, a shipping order is created, etc. If processing should be interrupted before all of these records are updated, the data base will be inconsistent. For example, inventory is diminished, but the order is never sent.

If the data base is not repaired quickly, the damage will spread. For example, an unnecessary order will be placed to replenish the inventory; inaccurate status information will be created. In batched data processing systems a large set of transactions are processed at once, and resulting data files used only if they are found to be consistent. For example, in a double-entry accounting system, a trial balance can be computed. If the accounts balance, no major processing errors have occurred.

Real-time data processing poses new design problems. In most cases, the transactions cannot be saved and processed in large batches. Cross-checking the data base for total consistency after each transaction makes the processing very inefficient.

With a transaction-oriented system the approach taken is to make the results of each transaction conditional, until the transaction completes successfully. If any processing problem arises between the beginning and end of the transaction, then the data base remains unchanged and consistent. When the transaction completes all of the updates are made as rapidly as possible, minimizing the time period during which the data base is inconsistent.

In a distributed system using more than one computer system, the systems must synchronize their update activity if some transactions update records on more than one system. Techniques for keeping multiple system data bases consistent are discussed in Chapter 13.

Information Transactions for Man/Machine Interfaces

Information transactions are an excellent way of packaging man-to-machine interchanges. Having a terminal operator interact with a computer system on a transaction-by-transaction basis has the following benefits.

- **Familiarity.** The style of interaction can be made very much like using paper forms, and quite free from internal computer details.
- **Controllability.** Specific transactions can be restricted to certain individuals or to specific terminal locations; information security is thus controllable.
- **Responsiveness.** Each transaction is assigned a processing priority, commensurate with the value of having it done promptly.

COMMUNICATION WITH MESSAGES OR COMMON DATA

In order to better understand the message-orientation of the model, let's examine a major alternative approach – a system-wide (or *dis-*

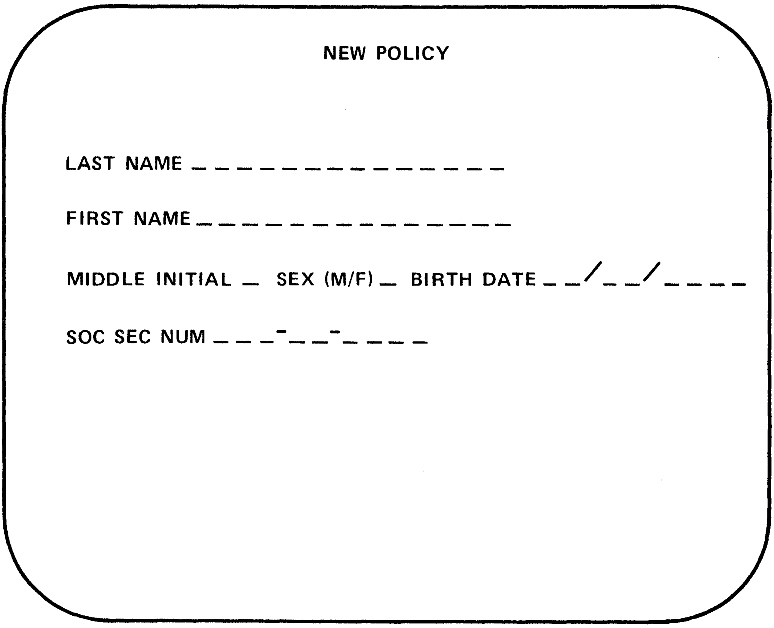


Figure 9-6. The use of form-based interaction is well-suited to a transaction-oriented system design.

tributed) data base. With a pure distributed data base system, programs are written to access data in the form of the enterprise-wide data base. This means that a program can be run on any system in the interconnected network (assuming direct electronic connection between system and data) without regard for the data storage location.

This approach is different from the work station model, where a program has an intimate relationship to the data and programs within the work station, but communicates with other work stations "at a distance" via messages.

Each approach has some advantages and liabilities.

Advantages of the Pure Enterprise-Wide Data Base

Complete flexibility. An effective automatic data base management system permits great flexibility in moving programs between systems, adding computers, moving computers, etc.

Consistent, uniform data management. With all enterprise data under uniform control, there can be minimal redundancy in the data base, and maximum ease in developing new programs.

Advantages in the Work Station Approach

Effective local use. Historically, large uniform data collections have become inflexible, because of the large amount of control needed to manage them. By partitioning the information system into work stations, the complexity of the individual data bases is greatly reduced; practical local usability is thereby increased.

Controlled responsiveness. The concept of work stations tends to force a good match between the location of data and its use. Being able to access a data item logically via an enterprise-wide data base is one thing; being able to access it quickly is another. Communications delays may be orders of magnitude greater than local file storage delays. Application programs that expect data access to take tenths of a second will not perform adequately when data access takes seconds.

Proximity to data. The ability to access a data item does not provide the ability to understand what it means. For example, six different manufacturing departments may each maintain an "inventory effectiveness" coefficient for each part, but there may be no obvious relationship between values from different departments. With the work station approach, there is an explicit need to consult with someone from another department when access is needed to data that is not available with an existing report. That consultation will presumably include discussion about what the data means. Given the total volume of data in the complete information system, formally defining each data item to the point where it could be used without person-to-person consultation is impractical.

Documentation of data is another proximity-related problem. From a practical point of view, it is unlikely that an enterprise-wide data base will be documented to a level that makes it fully usable to someone who does not understand the operation of the department that generates the data.

With work stations, the data presented in external reports is well documented but most data used internally is documented informally. The work station approach divides the data into a small portion that should be extensively documented and the majority, which can be informally described.

Value in information system planning. Specifying work stations forces managers and application designers to consider, up front, what the system should be. Having an enterprise-wide data base could be given as a reason for not having to do detailed planning. Practical experience indicates that such planning is essential to having effective systems.

Practicality. Specifying work stations forces explicit consideration of difficult questions:

- Why does this organization exist?
- What is the information flow?
- How will the organization change over time?

The results of this analysis can be factored into the design and can result in a cost-effective system. No distributed data base mechanism exists that can perform this optimization automatically.

Multi-vendor systems. The work station model permits a choice of hardware from different system vendors. Most systems can interchange character data via asynchronous communication and other existing protocols (e.g., 2780 emulation). Developments in packet communications services and standards (e.g., CCITT X.25) indicate that communication between systems will be enhanced in the next few years. In contrast, the chances of an effective distributed data management system, with different systems cooperating, are quite remote.

DISTRIBUTED DATA BASES IN A WORK STATION DESIGN

An automatic distributed data base mechanism is a valuable component of a distributed system, although it isn't a substitute for careful application analysis and work partitioning. A distributed data base mechanism permits data to be moved from one computer system to another without modifying the application programs that use that data. This flexibility is valuable in many situations.

Multiple Computer Work Stations

A distributed data base mechanism permits a single work station to be expanded by the addition of complete computer systems when additional processing capacity is needed.

Intrinsically Shared Data Bases

As described thus far, each data base is owned by a work station. Although this model represents the nature of most information within an enterprise, some data bases are commonly accessed by many work stations (a data base of corporate employee information with name, mailstop, and phone number, for example). Such a data base can be developed without a distributed data base by

1. Keeping copies at local work stations; these copies are updated periodically via communication messages from the work station with the master copy, or by

2. Creating a single "employee information" work station that responds to query messages from the other work stations.

The first approach may result in excessive duplication of storage resources and in the use of outdated data. The second approach may be highly inefficient for required bulk processing applications that use most of the data base, such as sending a notice to all employees or printing a phone book.

An automatic, distributed data base would be a desirable alternative in some cases.

Intrinsically Geographically Distributed Data Bases

Some data collections are necessarily viewed as a single data base, but must be geographically separated for efficient processing. An example is a gasoline credit card data base. The customer thinks of the card as being national or international in scope, but activity from a specific card holder comes from one locality for an extended period. Therefore it is cost-effective to store the records in a work station located near the active areas. This could be done within a pure work station model, but could easily benefit from distributed data management.

WORK STATIONS AND EXISTING COMPUTER SYSTEMS

For most enterprises considering distributed processing, a substantial data processing facility already exists. Do all existing application programs have to be rewritten as interactive work stations? If so, the conversion costs would make the work station model of little use. In fact, existing processing can continue and the model used as a basis for future development.

Work Stations as a Planning Model

The fantastic evolution in computer technology was largely unforeseeable. As a result, few long-term computer utilization plans are adequately aggressive. Now improved system cost/performance requires that more applications be automated. If long-term development plans are being revised to take advantage of the available power in distributed, interactive computers, the work station model is a useful planning tool.

Interconnecting Existing Systems

Any existing complete system *is* a complete work station. Even in a simple batch processing system, the input messages are jobs to be processed, and the output messages are the job results. By adding intersystem communication links, and making minor modifications to the job entry and printing functions, these systems can form a network of cooperating work stations.

Future Developments

Once a work station model has been adopted for planning and design, most applications are developed as work stations - interactive, transaction-oriented computer systems, with the processing aggregated into a number of work stations. As existing programs are improved and enhanced, they are made more like work stations. The net result is an information system that begins with what exists, and evolves over time into a network of work stations.

SPECIAL WORK STATIONS

The work station concept emphasizes the independence of the processing applications that constitute the distributed system. The emphasis on partitioning, and communication via messages, makes it easy to add special computer systems into the distributed system. The only requirement is that they be able to receive messages and send replies. The work stations don't all have to be transaction-oriented, on-line computer systems.

Manual Work Stations

It is valuable to understand important information functions as work stations, even if they are not automated. This will help in understanding existing systems and requirements and make future automation that much easier. Some of the work stations in the distributed system may be nothing more than an electronic mail drop – a manual department that receives messages from a teleprinter and later enters responses.

Control Work Stations

Control computers (process control, machine control, laboratory data collection) can be integrated into a distributed system as work stations. A small numerical control computer that operates a milling machine is a work station: it receives control program messages periodically, sets up the next job, and reports status information (e.g., parts finished count) back to the plant management system.

Query Stations

Transaction-oriented applications typically include query transactions for specific operational needs: display the customer list, display the orders for a specific customer, etc. Although these predefined transactions are simple and foolproof to use, they don't provide the flexibility of an interactive data query system, such as Digital Equipment Corporation's Datatrieve-11 PDP-11 system. Most distributed system data bases are an invaluable management asset when accessible via a convenient-to-use data query system which appears to the system as a slightly special work station.

Word Processing

Over time, more and more office typewriters are replaced by small, "word processing" computer systems, which enhance clerical productivity. If the word processing system can send and receive documents over communication lines (a capability of the Digital Equipment Corporation WPS-8 systems, for example), it can be integrated into a distributed system as a special work station that sends and receives documents.

Batch Processing Work Stations

For some work stations, batch data processing is efficient. Work may be done in batched mode (a stream of transactions processed at once) because

- the process is naturally done in batched mode (printing checks, doing year-end summaries) or
- processing many similar transactions at once is much more efficient, or
- batch processing improves data integrity (updating a master file off-line during off-hours, so that the updated data can be checked for consistency before it is placed in use).

In some cases a processing function is always done in a batched mode. In other cases, an application reacts to an exceptionally large workload by queuing some work requests for processing at a later time.

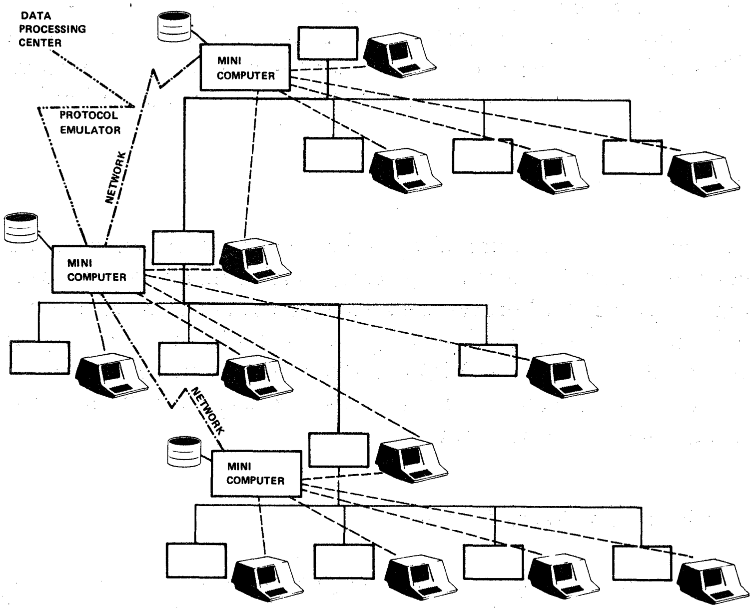


Figure 9-7. A network of interactive minicomputers is connected to a batch-oriented mainframe.

SUMMARY

Two important concepts in the design of effective distributed systems are the partitioning of the data base and applications into less-complex subsystems – work stations – and the partitioning of the processing into short yet complete pieces – transactions.

Chapter 10
DATA COMMUNICATIONS AND NETWORKS

Electronic data communications is increasingly important in computer-based information systems. Data communications systems are used to link computer terminals to remote computer systems, and to interconnect computer systems so that they function cooperatively. Data communications is a key part of distributed systems.

COMMUNICATING DATA

Definitions

"Communicating" means transmitting a message from one point to another. "Data," as we are using the term here, means information in a form that can be stored and retrieved in a computer system. In practical terms, data means a message that can be expressed in a textual form.

"Data communications" means moving computer-storable data from one point to another. Essentially any known form of communications can be used, as long as it can transmit reasonably complex messages.

For example, data from one computer system can be sent to another in several ways.

- by mailing a report, and having it reentered into the other system
- by dictating the data to another person, at the remote site, after which the data is reentered

- by mailing a magnetic tape from one site to another
- by using a direct, electronic communications link between the two computer systems

Characterizing Communications Methods

Suppose that we have a message consisting of several pages of text (10,000 characters). How can we characterize the differences between available forms of communications? The major attributes of communications schemes are the following:

1. **Cost.** What do we have to pay to have our message transmitted?
2. **Delay.** How long will it take the message to reach the destination? For electronic communications, delay means how long will it take from the time that use of a communication link is requested until message transmission *begins*?
3. **Capacity.** How big can the messages be? For electronic communications, capacity means "bandwidth" (How rapidly will the message be sent, once communications begins?).
4. **Reliability.** What percentage of the messages sent will be successfully delivered?
5. **Accuracy.** What percentage of the messages sent will be delivered without errors?

For example, suppose we sent the message as a letter.

> **Cost.** Using first class mail service, we pay the transmittal cost in postage. Other costs to consider include the cost for paper and envelope, the cost to write or type the letter, the costs of transporting the letter to and from the Postal Service, and the costs of reentering the data in the other system.

2. **Delay.** Within the United States, the letter would be delivered in 1 to 3 days, depending on distance, with some degree of unpredictability.

3. **Capacity.** Letter delivery charges are based on weight. Each additional increment of weight requires additional postage. There is an absolute maximum to the weight of a letter that may be sent. The amount of data that can fit into a letter is also a question of character size and spacing, weight of paper, etc.

4. **Reliability.** The reliability of the mail system is very high. We usually don't consider the possibility that a letter won't be delivered.

5. **Accuracy.** Some letters are delivered in a damaged condition (torn, water-damaged, etc.). There is some probability that the message won't be entirely readable. There is little or no probability that a different message will be received.

We could analyze, in similar fashion, the characteristics of any other message service.

Contrasting Electronic and Non-Electronic Methods

Why should we use electronic data communications rather than some other means? Let's consider electronic communications in terms of the same attributes. For the time being think of electronic communications as something to do with the telephone.

1. **Cost.** Electronic communications is rarely the least expensive means of data communications. As an absurd example, you might try computing the cost per character of sending a truckload of magnetic tapes (the data volume in a truck is very large; the cost per bit is very low).

2. **Delay.** Electronic communications almost always has less delay than other means of communications.

3. **Capacity.** Electronic communications rarely would be chosen on the basis of capacity. Again, as an absurd example, compute the capacity and bandwidth (the *rate* at which data is transmitted; the volume of data divided by the transmission time) of a truckload of magnetic tapes.

4. **Reliability.** The reliability of electronic communications is "good," but not outstanding compared to other means.

5. **Accuracy.** The accuracy of electronic communications is not particularly good. A relatively large percentage of messages are changed in transit.

Clearly the primary reason for choosing electronic communications must have something to do with delay.

Monologues and Dialogues

Delay is a factor in communications because too much delay makes dialogues impossible. Being able to hold a dialogue is often very important. For example, suppose we want to order a book. If we know everything about the book ahead of time, we can just send an order along with a check:

> Digital Press
> Education Services
> Digital Equipment Corporation
> Crosby Drive
> Bedford, Massachusetts 01730
>
> Enclosed please find a check for $20.95
> ($19.95 + 5% tax) for one copy of
>
> *Technical Aspects of Data Communication,*
> by John E. McNamara
> Digital Press, 1977
> Library of Congress Catalog Card No. 77-93590

Figure 10-1. One-way communication does the job if all the details are available beforehand.

But with much less specific information a dialogue may be used:

User: (dialing phone) "(617) 897-5111".

System: "Good morning, Digital Equipment Corporation."

User: "I would like to order a book,"

System: "I'll connect you with the Software Distribution Center."

(pause) "SDC, may I help you?"

User: "I would like to order McNamara's book on communications."

System: "Do you know what the order number is?"

User: "No."

System: "Is it used with the PDP-11?"

User: "No, no, it's not a program, it's a new book on computer communications!"

System: "Oh, yes, I read about that. I think that it's sold by Education Services. I'll connect you."

(pause) "Education Services, may I help you?"

User: "I want to order John McNamara's book on technical communications."

System: "Just a moment."

(pause) "This is Fred, can I help you."

User: "Yes, I want to order John McNamara's book on communications."

System: "Do you mean *Technical Aspects of Data Communication* from Digital Press?"

User: "Does McNamara have any others?"

System: "No, that's it!"

User: "OK, that must be it. What is the price, and where should I send the check?"

System: "Are you calling from Massachusetts?"

User: "Yes."

System: "Then the price is $19.95 plus $1 for tax."

User: "Fine, you'll receive the check and shipping address within a week."

Figure 10-2. Two-way communication has tremendous value if all the details are not available beforehand.

Imagine sending letter after letter, providing new information as each of these problems was solved. The advantages of a dialogue are obvious: you can work with initially incomplete data, rather than having to present exactly the right data in the right order; you can use the dialogue to work out the details.

Dialogues serve a similar purpose in electronic data communication. More and more data communication is via electronic means, and a primary reason is the ability to communicate in the form of a dialogue.

THE TELEPHONE SYSTEM

As interest in data communications grew, an obvious system to try to use was the telephone system. The telephone system is one of man's greatest engineering achievements. The system works so well that almost no one has any idea how much equipment comes into play in completing a call.

Transmitting Data Over the Telephone

The telephone system is optimized for human speech, and twenty years ago the only obvious way of interacting with the phone system was by making speech-like sounds. Simplistically, digital data is sent over speech facilities by having the digital data stream (sequences of binary one and zero digits, high and low voltages within a computer system) modulate a tone: a one causes a higher-frequency tone, and a zero digit a lower tone. At the other end of the phone link, a similar device "demodulates" the tone signal, and recreates a digital data stream. The devices that convert between voltages and tones are called modulator-demodulators, or modems.

The rate at which digital data can be sent over a communication link is limited by the bandwidth, accuracy, and noise properties of the link (noise is unwanted signals that are added by the communication system). The phone system is optimized for human speech. It transmits tones between about 300 Hz (Hertz, or cycles per second) and 3000 Hz. The accuracy and noise properties are controlled to permit speech to be understood at the other end. When a dial-up telephone line is used to transmit digital data, the signaling rate is typically 300 baud (bits per second) in each direction at the same time. At a signaling rate of 300 baud, it would take about six minutes to transmit our 10,000 character message.

Using Switched Connections. The simplest way to use the phone system is by dialing the connection. Specifying a phone number when placing a call results in the telephone system switching equipment selecting a set of linking circuits (the switches connect the circuits together) that create a circuit from the dialing phone to the dialed phone. The circuit can be used with telephones for a voice communication, or with modems for a digital communication. A dialed connection has many advantages.

1. *Flexibility.* As long as both ends have compatible modems, no particular telecommunications planning is needed.

2. *Cost-effectiveness.* The telecommunications facility charge is based on the duration of the call. By dialing the call when data communications is needed, good usage can be made of the service paid for.

But there are also significant disadvantages to the use of switched connections.

1. *Predictability.* The phone system is designed to complete a call by whatever means are possible when the call is made. There is no guarantee that a call will be completed with the same circuit on each call.

2. *Quality.* The quality of a specific call is particularly hard to predict. Not only will the circuits used vary, but the number of switches needed and the quality of the switching equipment will vary.

3. *Design Mismatch.* The telephone system was optimized for voice communications, not for data communications. For example, the telephone system is designed to provide adequate service based on the statistical properties of telephone call duration. Since most voice communications are of only a few minutes duration, lengthy data communications (a timesharing session, for example) can impact overall phone service by tying up facilities for much longer than is likely with a voice communication.

   The converse is also true: optimizations made on the basis of voice properties can impact digital communication. For example, when people talk, most conversations have one person talking at a given time, with slight pauses (fractions of a second) between one person talking and the next. The phone system makes use of this on some connections, and turns the two-way link into a one-way link, providing communications in only one direction at one time. This is a minor inconvenience for voice communications, but a total disaster for many data communications.

Using Dedicated Lines. In addition to providing dial-up facilities, the phone system will also lease the use of dedicated communications. A dedicated line is a permanant connection. Rather than using a different arbitrary switching path on each connection, the link is wired down. The advantage of a dedicated line is the predictable quality. Most of the problems introduced by the switching equipment are eliminated.

The disadvantage is that you lose the flexibility and cost-effectiveness of the dial-up line. The dedicated line goes only where you have planned, and the planning must be done some time in advance. And you pay for full-time use of the line.

With a dedicated line, the phone system provides additional signal quality guarantees at an additional cost (line conditioning). The quality of a conditioned line is not only predictable; it is predictably good.

Digital Telephone Links

Interestingly enough, inside the telephone system voice communications are transmitted in digital form to an ever increasing degree. The advantage of digital communication is that the impact of transmission noise can be minimized. With analog transmission, each leg (individual circuit) of the communication link adds some transmission noise, and if enough legs are used, or if some are particularly bad, the noise can overwhelm the signal, making communications impossible (when modems are used noise introduces errors in the digital data). In contrast, with digital links, after each leg the signal can be reconstructed and transmitted noise-free on the next leg.

For example, assume that a digital zero is represented by 0 Volts, and a digital 1 is represented by 1 Volt. If the received signal is 0.1 Volt we can guess that it was transmitted as a zero, and has been degraded by noise. When we retransmit the signal onto the next leg it will start as 0 Volts, free from noise once more.

A voice link is digitized by sampling it 8,000 times a second, and representing the analog value of each sample by an 8-bit data value. Therefore, to digitize our 300-3000 Hz analog voice link we create a 64,000 baud (bit per second) digital data stream.

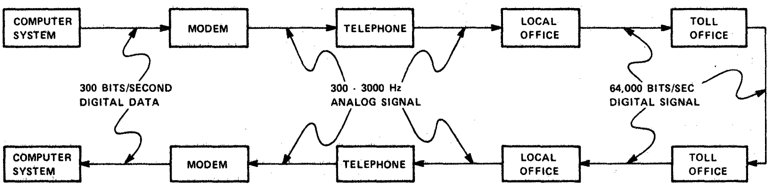


Figure 10-3. Inside the telephone system digital data communication is used to transmit voice signals. Using modems to convert digital data to a voice-like signal leads to very inefficient use of these digital links.

But suppose we are using the voice link for digital communication with a modem at 300 baud. We use the modem to create a 300-3000 Hz analog signal, and now we've digitized that signal into a 64,000 baud data stream. To say the least, something has been lost in the translation! If we intended from the beginning to do digital communication, we are now using about .5% of the digital capacity inside the phone sysem! Not only does that make the link cost more, but if we could just gain access to the 64,000 baud link in the first place we could send our 10,000-character message in just over one second, rather than the six minutes it takes with the 300 baud link.

If we could use these internal digital links our utilization of the investment in cables, amplifiers, microwave links, and the like would be greatly increased. But we would still be left with the problems of flexibility (the normal dial facilities don't directly access the digital links) and reliability (what do we do if *our* link stops working?).

The end solution is to add computers, and to create an intelligent digital communication system.

INTELLIGENT COMMUNICATIONS AND NETWORKS

The telephone system consists of various forms of communication links, and switches that connect the links to form end-to-end communications links for users of the service. To make an efficient data communication system we will use many of the same communication links (especially the high-performance digital links), but we replace most of the analog switches with dedicated computer systems.

Until recently, telephone switches were large complexes of electromechanical equipment. Given the largely mechanical technology, the speed and reliability of telephone switches is quite remarkable. But modern switches are essentially specialized computers. The incoming analog signals are digitized (this function can now be done by a few ICs) and the digitized values are "switched" by the computer, which moves them from input registers to output registers to complete the call.

Digital telephone switches are much more flexible than analog switches. Since the switching is done by programs, rather than mechanical switches and cables, the details can be easily changed. Because of the low cost of computers and memories, some of the new switches have enough capacity to permit all incoming lines to make calls at the same time (whereas mechanical switches provide only enough capacity for marginal service under peak load).

All in all, the move toward digital telephone switches makes designing data communications systems much easier. For the present, special-purpose digital switches, dedicated to data communication, exist in addition to voice-service phone switches. In the future it's quite likely that a single telephone system will provide both voice and digital service perfectly well; such service will eliminate much of the need for most specialized digital communications systems. A single digitally-based telephone system will serve many functions that now use various specialized facilities: facsimile transmission, teletypewriter, news services, stock ticker tapes, electronic mail delivery, telegrams, etc.

Distributed Systems Objectives

Distributed systems use a digital communication facility for particular functions:

- to interconnect terminals and computer systems
- to send messages from one work station to another
- to provide some form of distributed data access

These functions suggest objectives for a data communications system:

1. **Accuracy.** We want accurate data transmission. Communication links are subject to all sorts of temporary disorder, such as lightning storms, which introduce errors in the communicated data. For the most part we aren't interested in dealing with these errors. If possible, the communication system should make errors invisible to the user of the service.

2. **Efficiency.** We would like a system that makes good use of the investment in facilities. We should pay for only the service we use.

3. **Simplicity.** We would like the system to be as easy to use as the dial phone system. If we want to talk to the "accounts receivable" work station we should be able to address it symbolically, rather than by having to know about the "4th line on the B multiplexer".

4. **Reliability.** We want communication to go through if it is reasonably possible, just as the dial-up system places the call in any way possible. The fact that a microwave tower in Montana has just been struck by lightning, reducing the available

links across the area by 1%, shouldn't mean a catastrophe for us, just because our dedicated line uses that facility.

5. **Functionality.** If the system is implemented by computers, we might as well ask for nice extras. For example, we could ask for the capability to broadcast a message to many destinations at once. We could even ask that the communication service translate data from one form to another, for example, COBOL representations to FORTRAN representations.

6. **Universality.** We would really like one standard scheme that would work everywhere, in all cases. In the ideal, we could construct complex systems using many different computer models, operating systems, and programming languages, and yet have the application programs able to communicate with one another.

Models for Data Communication Systems

Good, standard conceptual models do exist for data communications systems. Although the details of the many systems vary widely, most fall within the conceptual bounds of the model presented here. The use of the model makes it easier to explain what the different parts of a communication system do and to relate the features provided by one system to those provided by another.

Before we introduce the model, let's briefly summarize what we are trying to do:

1. Provide convenient means for data processing programs to communicate data between themselves in computer systems that may be geographically dispersed.

2. Use existing communications technology, such as services provided by the telephone system.

3. Provide for the known properties of the communications technology; in particular, recognize that the communications links make errors.

4. Provide cost-effective services.

The model consists of a number of functional layers. The layers begin at the "bottom" with physical communication links and end at the "top" with programs. Each layer solves one set of problems, and presents a higher-level function to the next layer.

A layer is a "black box" that provides the specified function. It may be implemented in software on a general purpose computer, in a specialized computer, or in specially designed hardware.

A specific implementation isn't necessarily designed with exactly these layers: two layers described here may be one or three in a specific architecture.

The foremost reason for defining specific layers is to establish interfaces that can be standardized. If we ever hope to achieve our objective of universality, the black boxes developed by different vendors must work together. This requires some level of standardization. The functional layers of a model are an obvious place to standardize.

These are the functional levels:

1. **Physical Link.** The communications medium.

2. **Logical Link.** The physical link, adapted to provide such functions as error control and multi-drop addressing.

3. **Logical Channel.** A logical data pathway between two communicating programs. Many logical channels may be served by the same logical link.

4. **Communications Session.** A complete dialogue between two programs, including finding the other party, initiating the dialogue, transmitting data, and terminating the dialogue.

5. **Data Presentation.** Translating data between the data representation formats used by two different programs (integer data in FORTRAN binary-representation and in COBOL decimal-representation, for example).

6. **Application Program.** Finally we have the program that needs to communicate data.

Physical Link

Each link is some form of communication line, such as a high-speed digital phone link. Each link has key characteristics.

- **It makes errors.** The error rate tends to change over time, rather than being statistically random as is the case with digital logic on a disk drive. The errors tend to occur in bursts.

- **It has limited capacity.** Only so many bits per second can flow over the link. High error rates will further limit the productive throughput.

- **It has a transmission delay.** There is a delay between the transmission of a bit and its receipt. The delay may vary from millionths of a second, for short wires, to significant fractions of a second for a communication satellite.

- **It will sometimes fail totally.**

Logical Link

A logical link is an adaptation of a physical link whose purpose is to conceal as many of the problems as possible. For example, a logical link is error free. Additionally a logical link can have many "ports" that can be viewed like on and off ramps from a highway. Whereas a simple electrical circuit has a beginning and an end, a logical link can be used to communicate data between any of the ports.

Error Control. There is no way to make the physical link error free. Transmission errors will continue to occur. Our objective in building a logical link mechanism is to buffer programs from those errors, in the same sense that a secretary discovers what the memo writer really meant and corrects obvious errors.

There are two basic ways of handling errors: redundant transmission and retransmission. In a redundant transmission system, the original data is converted into a form that includes a level of redundancy. Then, if part of the transmission is damaged, there will usually be enough good data left to reconstruct the original message. An example of redundant transmission is modern high-density (6200 bpi) tape drives. The technique is particularly valuable for tapes because of high mechanical overhead in stopping the tape, reversing the direction of movement, reversing it once more, and rereading the data.

Error Control by Retransmission. For communications links retransmission is a more common scheme. With retransmission we simply cause blocks with transmission errors to be transmitted again, until they are received correctly. For any particular error rate on the physical link, the block error rate can be reduced to an arbitrary level by making the blocks shorter.

A basic problem is how to determine when transmission errors have occurred. This is done by adding to the data a code that has been computed from the data contents. Then if either the data or the code has been altered, the code will not match the data on receipt.

Multi-Drop Connections. If we want to have multiple ports on a single physical link (like multiple outlets on a home electrical circuit) we can easily provide for them in the design of the logical link mechanism. Each block of data transmitted can have an addressing header that specifies which port the data is intended for, and which port sent the data.

Transmission Delays. We can also make most of the effects of transmission delays disappear. Each logical link port is implemented by a small computer that attaches to the physical link. Programs executing in the computer transmit blocks to other ports, receive blocks from other ports, check data transmissions for errors, etc. If the port computer system has enough memory to buffer multiple data blocks, we can keep the physical link "pipe" full and at least not lose any throughput because of the delay in transmission.

For example, we need to hold a block at the transmitting end until we are sure it has been received correctly. In the simplest scheme we would send one block, and then wait patiently for a response saying it was received correctly. If we didn't get the response in a reasonable time, we would retransmit the block (each block is given some sort of unique sequence number, so it doesn't hurt to receive the same block twice). The problem with this scheme is that if the delay was one-half second each way, and we send relatively short blocks of data, we would be spending lots of time waiting, and very little actually using the physical link to transmit data.

But if we give the system implementing the logical link enough buffer memory, it can continue to send blocks down the "pipe" without waiting for a response on a block-by-block basis. In other words, we can send ten blocks without receiving the acknowledgment that the first was received correctly, and continue to transmit new blocks, until we run out of buffer memory.

Unresolved Issues. We are still left with the problem that the physical link may develop a very high error rate, or fail entirely. And we still have only a finite capacity on the link. In fact, we have implemented logical links by using some of the data communications capacity for sending control and status information back and forth (error detection codes, addressing headers, receipt acknowledgment), leaving less capacity for data transmission.

Logical Channel

A logical channel is created by a set of programs using a logical link. A logical link lets us use a physical link, with multiple ports, and delivers blocks (or packets) of error-free data. So far we have made some progress in achieving our objectives of accuracy and efficiency. But many problems remain. For example, what do we do if the programs that wish to communicate with one another aren't both on the same physical link?

The next conceptual level is the logical channel, which provides a data pathway between any two programs that can communicate. It addresses the following problems:

1. Finding the other program
2. Communicating by using multiple logical links

3. Sharing facilities with other programs
4. Using physical resources efficiently
5. Adapting to changing load
6. Adapting to failure

Simplistically, the logical channel is a set of programs that permits one program to send an arbitrarily long message to another by

1. Finding where in the network the other program is
2. Breaking up the message into smaller pieces
3. Transmitting the pieces as data packets, over logical links
4. Choosing the links on the basis of message priority, availablility, and other traffic
5. Reassembling the message at the other end, in the proper order, with all the packets correctly transmitted

Finding the other program. A communication system has various addressing and naming problems. At the bottom conceptual level – the communication hardware – the control program must select specific communication lines by specifying actions on particular devices. For example, the phone line to the Chicago sales office is attached to a modem connected output line #12 on device controller #4.

At the top conceptual level are the application programs. If I am writing an Order Processing application, I may wish to send data to the "Accounts Receivable" application. In my application program I would like very much to refer to the other program in a symbolic manner ("ACCOUNTS RECEIVABLE") rather than having to know on which system it is running, and which combination of other systems, physical links, and communications device controllers are needed to effect the transmission of data.

The function that implements the logical channel includes the capability of finding a program (message destination) by some symbolic name, and providing the detailed information needed to effect communication. This does not happen by magic. There need to be mapping tables and programs to create, maintain, and use the tables. But we definitely want to be able to move programs, and add, modify, and delete physical links, without having to modify the details of each application program using the communication services.

Communicating by using multiple logical links. If two systems are not connected directly by a physical link, then we need to use an intermediate link as a stepping stone. Simplistically, a data routing scheme can work as follows:

1. The originator, System A, wants to send data to System B. A control table in System A states that the way to send data to B is over physical link 1, which does go to System C. The table would also show link 1 as the way to send data to System C.

2. System C receives a data block over link 1. After verifying that the data has been received correctly, it examines the addressing header. The header shows the data is for System B.

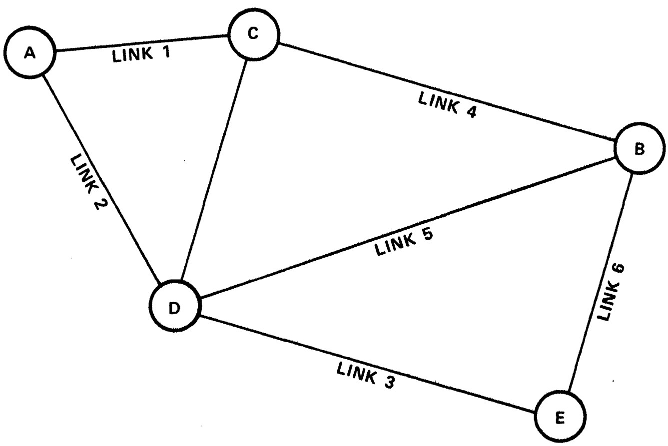


Figure 10-4.

3. A control table in System C shows the best way to send data to System B is via link 4, which happens to go to System B.

4. System B receives a data block over link 4. After verifying that the data has been received correctly, System B sees that the block is addressed to itself, and uses further header data and control information to route the block to the recipient program.

Sharing facilities with other programs. In most applications we can assume there will be many programs communicating data, for different purposes. The logical channel shares the limited physical resources by multiplexing many different messages over a single physical link. There is no reason why all the packets of one message have to be sent in sequence. The logical channel can send one packet from one message, two packets from a second, one from the first, etc.

We can also assume that in terms of the application, some functions are more important than others, and should receive priority in the use of resources. The logical channel can implement these priorities by the way in which it selects which block to transmit next. The logical channel can allocate one fraction of the resource to one message and a much larger fraction to a more important message.

Using physical resources efficiently. With the ability to route data packets flexibly, which we have introduced to be able to communicate with a program via intermediate systems, we can also get around another of the physical link limitations, namely a bounded capacity. If there are two independent pathways from one system to another (an assumption), we can use both to transmit a multiple packet message. The phone system will attempt to place a call by a very indirect route, if that is the best way available when the call is made. Analogously, we may choose to use indirect data routing, if it increases the total system throughput.

Adapting to changing load. Given that we have some programmable control over routing, we can change the specific rules for transmitting messages from one system to another on the basis of load.

For example, if a heavy load develops between System C and System B then it may be better to route messages from System A to System B via System D. By changing the data routing rules we can continue to deliver service in the priorities set by the application, while using whatever physical link capacity exists.

Adapting to failure. Although multiple routings between systems permit us to optimize data flow, the primary advantage of multiple routings is being able to endure the failure of any single physical link. When a link fails, we can re-establish routing rules so that messages continue to flow around the failure.

Progress against objectives. By adding functions that implement logical channels we have made great progress against our initial objectives.

1. *Accuracy.* Not only do single blocks transmit correctly, but we can get around all sorts of other problems like losing physical links. Now messages have a high probability of transmitting accurately.

2. *Efficiency.* We can share physical links among programs, and even use multiple links for a single message. We are able to make good use of the hardware resources.

3. *Simplicity.* We can address message recipients symbolically. Further simplifications are achieved at the higher levels of the communication facility.

4. *Reliability.* We have added many features that improve reliability.

5. *Functionality.* We haven't done much yet except add basic functions.

6. *Universality.* We don't really know. We have invented a logical scheme that works on the systems for which we have implemented it. It achieves universality if there are many systems that use the logical scheme.

Communications Session

What we have theoretically built so far permits actively running programs to exchange messages. That is a big step forward, but doesn't exactly meet our initial needs. We wanted to be able to send a message to the "ACCOUNTS RECEIVABLE" program. But what if that program isn't running at this instant (because it has nothing to do), or what if it isn't expecting a message from my program (do we have to agree that the message will occur at exactly 3 PM each day?).

These problems and others are overcome by creating the concept of a communications session, and providing programs that manage these sessions. During the sessions, data is transmitted via logical channels. The session logic is primarily concerned with the problems of starting all the right programs running when needed and establishing the logical channels between them.

With this more sophisticated view of communications, we would expect actual communication between programs to have the following general steps.

1. One program initiates the communication by requesting a session with the other.

2. If the request succeeds, the other program is then ready to receive message data and a logical channel has been set up between the two programs.

3. Messages are sent and received. They are broken into packets and transmitted via various physical links.

4. One of the programs requests that the session be terminated.

5. The logical channel is relinquished, and the program resources that exist for the communication (e.g., buffer space) are released.

The kind of functions that can be implemented in the session controller include the following:

1. **Initiating program execution when a message is received.** The communications subsystem may cooperate with the job scheduling subsystem to cause a particular job to be initiated when a particular session is requested. For example, a request for an "ACCOUNTS RECEIVABLE" session causes the task image file ACTSRECB.EXE to be executed.

2. **Communicating session requests to an executing program.** By means of a program interrupt or exception facility, a running application program may be notified that a request to communicate with it has been received from the network.

3. **Queuing incoming messages.** For some applications we might wish not to process received messages immediately, but rather queue them locally on a data file and process them at a later time. This function could be implemented as a form of session.

4. **Providing data packaging services.** In some circumstances, a single message may have no meaning without further messages. For example, messages are received from a process control computer. Six messages are sent in sequence, each representing different values of sensor readings. The messages are sent individually because of limited buffering space in the process control system, but the data has no real value until we receive all sensor values. We may provide session control capabilities that "know" about the structure of messages, and only activate the receiving program when the complete package has been received.

5. **Work synchronization.** A problem that is discussed in length on the chapter in design techniques (Chapter 13) is keeping data consistent. In order to assure that the data bases on multiple systems are consistent, we need to be able to synchronize the application work performed on multiple systems. For example, a transaction that updates data on four systems will

not succeed until all four updates have been done correctly. If any one of them fails, then the other three must be nullified to keep the complete data base consistent.

Data Presentation

We now have a rich set of functions for message interchange. The final service that we would like to layer on top of these is data presentation services: converting data between "network standard" formats, and local system formats. We will consider here two examples of data presentation:

1. Data type conversion

2. Terminal dialogue conversion

Data type conversion. Suppose we have two programs that wish to interchange integer data, but one program is written in FORTRAN, and represents integers formatted as 32-bit binary quantities, whereas the other is a COBOL program, which represents an integer formatted as a variable-length, packed-decimal string. If we send the bits representing a binary integer to the COBOL program it will certainly not be the right sequence of bits for the packed-decimal representation of the same integer.

What we want is to have a set of conventions for transmitting various forms of data on the network, and a set of subroutines to convert these formats to the internal formats used in the various systems.

The concept of data type conversion can be extended to the reformatting of records. Two systems may each have an employee file, with the same data items in each employee's record, but with the items in different order. A presentation service could reorder the data fields while transmitting the records between the systems (much like the function performed by the COBOL language MOVE CORRESPONDING verb), as well as converting between different internal representation formats.

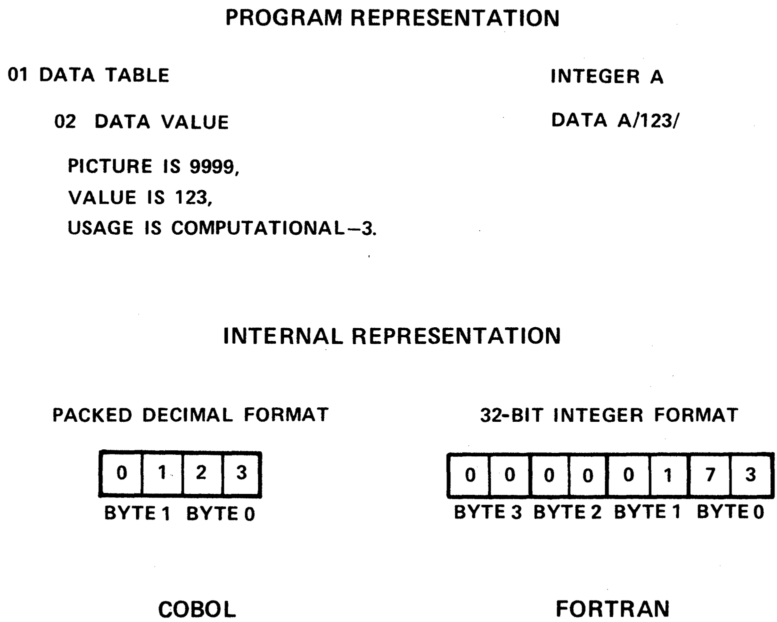


Figure 10-5. Data presentation services in a network convert integer data between the internal form used by a COBOL program and the internal form used by a FORTRAN program.

Terminal dialogue conversion. A common type of conversion involves interfacing a program to a terminal user. Suppose we think of an order entry as primarily consisting of message preparation by the terminal user. The receiving program, which executes the order entry transaction, would like to receive this message in a well-defined, well-structured format. In contrast, the terminal user prefers to enter the data needed for the message by a dialogue process:

1. A sales entry form is presented on the display screen.

2. The terminal user enters data for some of the blanks on the form.

3. The user is told when data fields that have been omitted are necessary (e.g., account number is mandatory, whereas customer phone number may be optional).

4. The user is told when data fields are obviously wrong (e.g., account number check digit is wrong, letters appear in the ZIP code of the billing address, etc.).

5. The user corrects the omissions errors detected in #3 and #4.

6. Finally, all of the extra formatting and explanatory text is stripped from the input data, and a concise fixed-format data message is sent to the program.

In the case of data being reported back to the terminal user, a similar process can be imagined in which the terse, fixed format message from the program is expanded for the terminal user.

These conversions between program-optimized messages (terse, fixed-form) and user-optimized messages (dialogues) can be viewed as a specialized kind of data presentation service.

UTILIZING COMMUNICATION SERVICES

We can use the communication services that we have specified thus far to send messages back and forth between cooperating programs, and between programs and message-oriented terminals. We can also use these services, with a small amount of additional software, to implement other useful system functions:

- "Virtual" terminals
- Resource sharing
- File transfer
- Remote file access
- Distributed data base

"Virtual" Terminals

The concept of a virtual terminal is very simple. Consider just System A for the moment, and assume that System A is an interactive system that is normally accessed via terminals. Now that we have developed a communications network, we would like to use a terminal, which happens to be connected to System C, as if it were connected directly to System A. In other words, after some initial dialogue with System C, during which we request a virtual attachment to System A, the terminal should behave exactly as if we were attached directly to System A.

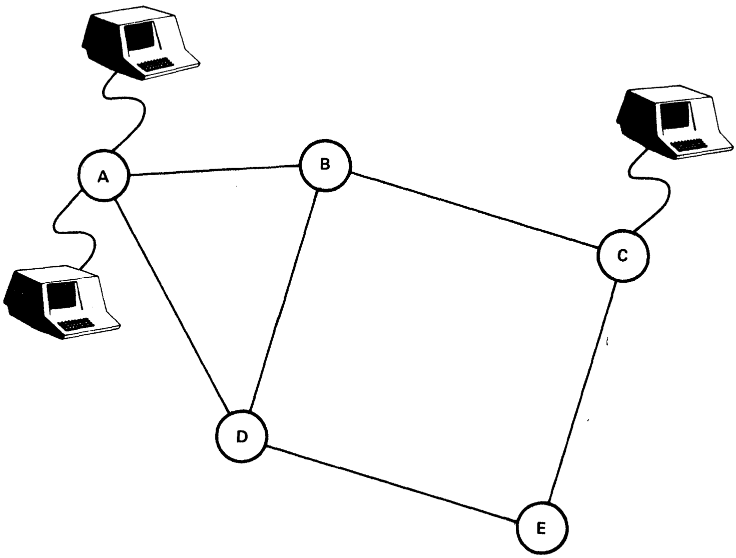


Figure 10-6. Virtual terminal communication services permit a terminal on System C to be used as if it were directly connected to System A.

The problems encountered in virtual terminal support relate to how much of the terminal interaction depends on an intimate, responsive connection to the host system. If the system is designed so that a terminal sends one line of data at a time, then the problems in implementing virtual terminal support are small. But if the system depends on character-by-character interaction, providing virtual terminal support is more complicated.

For example, assume that the system normally does character echoing to the terminal and provides simple line editing. Character echoing means that the terminal keyboard and printer or display are operated as independent devices, as far as the system is concerned, with full duplex communication (active in both directions at once). If I strike the "A" key on the keyboard a code is sent to the system, which in most cases then sends the "A" code back to the display or printer. This permits the system to implement various forms of functions not directly implemented in the terminal. For example, the system may provide simple line-editing functions, such as erase the last character, erase the whole line, show the line as it stands now, etc.

If the terminal is now linked via a network, and the individual characters are sent as single-character messages through a network, there will be two disastrous results:

1. The responsiveness of the terminal will be very poor. Echoing a character a few tenths of a second after it is typed has little or no perceivable impact on the terminal user. Echoing it a half second later is very disconcerting to the typist.

2. The communications facilities will be clogged by very inefficient one-character messages (one character of data and tens of characters of header and routing information, checksums, etc.).

In most cases, the simple approach of sending terminal input character-by-character using the communication network is not practical. But what can be done in most cases is to identify a large subset of useful interactive functions, such as line editing, and implement them uniformly in each system. Then the systems can provide responsiveness locally, and use the network to ship completed messages, such as whole lines. This approach preserves terminal responsiveness, and makes much better use of the communications facilities.

Resource Sharing

With communication services installed, we may choose to share some expensive or infrequently used peripheral devices among systems. For example, we may install a single high-speed printer, or microfilm writer, and use it automatically throughout the system. As the cost of computer systems and other electronic equipment continues to decrease, it becomes more and more attractive to share expensive electro-mechanical peripheral devices.

Resource sharing can be implemented by a set of compatible programs on each system in a network. The programs can interchange messages in a common format, and by a common set of rules, and know the local conventions for accessing the peripherals.

File Transfer

An obvious facility to build is a utility that conveniently copies data files (parts of the system data base) from one system to another. We can use this utility for resource sharing by having one system invested with expensive, but cost-effective large disk storage units. Intersystem file transfers can increase data reliability by periodically sending copies of all updated files to an archive system where we take special precautions (such as using fire-resistant storage vaults) to protect the data.

The simplest file transfer utility may just send binary data images from one system to another (transfer a file as a sequence of bits). More sophisticated utilities may do translations from one format to another. For example, a character file (textual form data) could be translated from the character set of one computer system to the character set of another. Or an indexed record file could be transferred as records, and rebuilt into the indexed format of a different system.

File transfer can be implemented by a set of programs that interchange messages, and know about local file access and formatting.

Remote File Access

The next increase in sophistication beyond file transfer is the ability to access files located on a remote system. Normally a file is accessed by "opening" it and then performing a set of positioning, reading, and writing operations on it. Opening it consists of interpreting the name of the file, and determining where the data is stored.

In the case of remote file access, opening a file consists of

1. Determining which system had the needed file

2. Establishing a communications session with a cooperating program (a file transfer utility) on that system

3. Performing an open function for the file on the remote system

Having opened the file, the program writing or reading the file issues standard file-access commands (e.g., read, write, update). Instead of executing these commands, the system on which the program is executing transmits these commands to the system at which the data was located, via the communications network. System-provided utility programs access the remote file as requested, using the communication network to transfer data between file and program.

Remote file access can easily be designed so that a file can be moved from one system to another without changing the programs using the file. Designing effective, interactive data processing programs is another matter: the delays introduced by moving a data file to another system and accessing data in the file over a communication network can easily slow the program response to the point where the program is no longer practically usable. If the files are accessed by more than one program at once, the design problems increase because of the additional delays needed to lock records, synchronize updates, etc.

Distributed Database Management

Database management systems differ from the file management systems we have been describing so far in many ways. In particular, when using Database management services, the application programmer has no concern with the physical management of the data (how the data is organized physically as opposed to logically, etc.). The application programmer deals strictly with a logical view of the data. Most Database systems even permit two programmers to be given different views of the same data records; specific data fields may be made invisible.

Extending Database concepts to a distributed systems environment means that the programmer is not even concerned with the system location of particular data items. Requests to the Database system for data that is not locally resident causes messages to be sent among the Database manager programs in the network, and ultimately results in the data being provided to the application requesting it.

It's tempting to think of distributed Database access as *the* mechanism for building distributed applications. You just write the programs without any regard to location and then create the global Database. We present another view, that distributed systems are best designed with some explicit partitioning of the whole system into functional subsystems, called work stations. Nonetheless, being able to do some distributed Database management will be a valuable addition to systems of the future.

PROVIDING COMMUNICATION SERVICES

For any given computer application, there are three approaches for providing communication services:

1. Design and implement unique services for this application.
2. Use services provided by a communications service vendor.
3. Use services provided by a computer system vendor, or a system house.

Designing Unique Services

This is always the avenue of last resort. If there is no other way to obtain adequate services (for example, very-high bandwidth communications), they can always be explicitly programmed. The disadvantages are many.

1. **Cost.** This is likely to be the most costly approach.

2. **Efficiency.** Because of the difficulty of creating these services, it is quite possible that the performance will not meet initial expectations without additional investment.

3. **Compatibility.** It is unlikely that the system will work with other systems, unless compatibility is an explicit design goal.

The Telephone Company

In the United States, the telephone company has provided many types of efficient communication services, such as superb voice service, teleprinter message service, broadcast transmission services, etc. They have, however, been relatively slow to develop digital communication services. There are three major reasons for this:

1. Like everyone else, the phone company failed to anticipate the phenomenal growth in computer usage.

2. The phone company is encouraged to move slowly by the legislation under which they are regulated. Service rates are set to amortize capital expenditures over a lengthy period. Technically advanced equipment cannot be introduced until the existing equipment is paid for, by which time it is obsolete. (The legislation was originally designed to regulate railroads, where technological progress is slower.)

3. Because of their monopoly position in telecommunications, the phone system has been restrained from branching out into other businesses where their monopoly might give them unfair competitive advantage. Historically, they have been explicitly enjoined from offering "data processing" services.

However any clear division between the telephone industry and the computer industry is breaking down rapidly.

1. The phone system has been permitted to rent intelligent terminals to their customers, despite the complaint that they are essentially small computers, and in that way provide data processing services. Many feel that this decision assures that the phone system will be able to provide more and more computer-based services. The existence of low cost computers, and the extensive use of computers within the phone system itself makes it harder and harder to say a service can't be provided because it uses computers.

2. Telephone communications use more and more digital techniques. Essentially all new telephone switching systems are designed with computer-logic technology.

3. Large communications customers are looking for integrated solutions to their data and voice communications problems. APBXs (automatic switchboards), which provide for both telephone and terminal switching, are now available.

Telephone Standards - CCITT

Outside the United States, there is less legislation preventing a phone company from providing other services. In fact, in many countries the phone system is run by the government. Many of these phone companies look at the move to increased data communications as a way to earn a larger share of the total information processing revenues (the computer business has been dominated by United States companies).

The phone companies have an international standardization body, CCITT (Comite Consultatif Internationale de Telegraphie Telephonie), which functions under the aegis of the United Nations. CCITT has taken an aggressive stand toward developing data communications standards. The CCITT X.25 standard has been a leading force in the development of network standards.

The international telephone companies have an obvious incentive to develop higher-level data communications standards, and to provide economically attractive services. Since the phone companies are experienced in developing cooperative standards for international phone service, CCITT may prove to be a leading force in standardization development.

Communication Vendor Services

Because of the slowness of the phone company in the United States to develop data communication services, various specialized service companies have been formed. These companies buy communications facilities wholesale from the phone system (or in some cases develop their own) and add specialized interfaces (often communications computers) to provide various retail communications services to their customers. Because they operate like distributers, some have been called "value added" networks.

The services provided are still relatively crude, compared to the full functionality we have defined. In general, it is necessary to develop special system software to interface to the network. As time goes on, these service companies will certainly develop more extensive software so they can expand their markets to a greater number of computer system users.

Computer System Vendor Software

Another source of communication services for distributed systems is computer system vendors. With this approach, customers purchase relatively primitive communications services and the necessary software, and then implement most of the functions on their own computer systems. Essentially every major computer system vendor has jumped on the "distributed processing" bandwagon, and announced "network" software of one form or another.

Caveat Emptor! The goal of the computer system vendor is to sell computer systems – as many as possible as soon as possible. This goal may or may not be compatible with a customer's needs in distributed systems, which are long lasting and have a fundamental impact on the ability of the enterprise to perform efficiently.

Although the phone system is also interested in making money, there are some interesting contrasts between the phone system and a computer vendor. For one, there is little chance that the phone system will become financially insolvent, or choose not to sell phone services any more (how many former computer vendors are now making video tape recorders, or copiers, but no longer making computers!).

There is also little chance that the phone system will emphasize short term profits excessively, since the legislation under which they operate requires a long term fiscal view. The phone system usually sells only services that are totally over-engineered by any other commercial standard, and that can be used immediately and reliably.

In these respects, the specialized communications vendors are more like computer system vendors than like the phone company. Many have tried but few have succeeded. A specialized service vendor may supply expertise in difficult technology, but may not provide a reliable service over the long haul.

Some critical questions in assessing the offerings of computer system vendors include the following:

1. **Design Quality.** Is the network well designed and tested? It is one thing to print a four-color brochure offering network services, and another thing entirely to make it work.

2. **Generality.** Do the network capabilities provided by the vendor permit flexible use of the various hardware and software products? Can you select the local computer systems that constitute the distributed system on the basis of local needs, or does the network offering restrict which systems can be used?

3. **Proprietary Design.** Can you obtain a detailed design specification? Can you build compatible software on hardware supplied by another vendor? Do you have some idea about how the network design will evolve in the future?

4. **Need for Special Hardware.** Does use of the network software require scrapping an investment in hardware such as modem equipment? Can inexpensive hardware be used where performance is not critical? Does use of the network require special terminals, or can conventional terminals be used?

5. **Serviceability.** When you make an investment in a network, will the vendor be able to keep the network working reliably? Does the vendor have adequate hardware and software service organizations? What is the history of the vendor's dealings with other system vendors, and with communications suppliers?

Being dependent on a computer system vendor is neither new nor necessarily bad. Anyone who buys a computer system is somewhat dependent on the vendor to produce quality hardware and software and keep it working. But moving to distributed systems deepens the dependence. As computers are used more widely, they become a more critical part of the basic operation of an enterprise. A single laboratory computer failure typically has much less impact than having the entire front office of a business cease operation.

STANDARDIZATION EFFORTS

Distributed systems require standards. Two (or more) systems will communicate easily only if they abide by the same rules. The rules can be created by the system developer, the computer vendor, or some other standardization body.

The computer industry, with the prodding of powerful customers, such as the United States Government, has made some progress in standardization, at least at the level of character sets and common programming languages. But the standardization mechanism is neither timely nor highly responsive to the immediate needs in the marketplace.

Why Worry About Standards?

Many enterprises would prefer multi-vendor computer systems:

- The existence of multiple vendors leads to choice in purchasing, which keeps vendors highly competitive. Competition spurs on technological progress and assures reasonable pricing.

- With multiple vendors providing functionally comparable systems, the future of the enterprise is decoupled from the future of one particular system vendor.

Standards are one way of assuring multi-vendor options. In the semiconductor industry, there is voluntary "second-sourcing" in which the producer of a proprietary product often actively seeks another company that is willing to produce the identical part. Often the original producer will supply the second source with detailed design and fabrication information. In the case of semiconductors, large segments of the marketplace are unwilling to use single-vendor parts, for fear that any kind of vendor problem, including limited capacity, will prevent the part user from building his own equipment. So far, second sourcing has not been developed in the computer industry, except for companies wishing to exploit compatibility with hugely successful products, although there are some indications that the Japanese computer industry is developing some kinds of system-level second sourcing.

Standards also will play an important role in distributed systems because of the use of communications facilities, an area that is already subject to standardization and regulation. Communications systems have been standardized for many reasons:

1. To permit complex systems to be built in the first place. Often these systems consist of parts from multiple vendors, or even different operating companies (e.g., independent phone companies, AT&T Long Lines).

2. To define standard interfaces and levels of service, so that tariffs could be established by the regulating body.

3. To permit systems to be built across national borders.

Distributed Systems Standardization

The need for distributed system standards is evident. Computer vendors have shown active interest in the standardization effort thus far. Standardized programming languages are obviously a good idea, but hardly critical for accomplishing work. There are many computer shops with multiple different systems without absolute standardization of language between them, and yet work goes on. But there are no distributed systems without standards for interconnection and transfer of data.

Because of the perceived needs, the International Standardization Organization (ISO) Technical Committee 97 (data processing) created Study Group 16 (Open Systems Interconnection) to examine this area. ISO/TC97/SG16 held its first meeting in Washington, D.C., in early 1978. To prepare a United States position, the American National Standards Institute (ANSI) Standards Planning and Research Committee (SPARC) created Project 300 (Distributed Systems – DISY). DISY is a study group consisting of expert representatives from computer vendors (e.g., IBM, UNIVAC, Burroughs, Honeywell, DIGITAL), vendors of asssociated products and services (e.g., Boeing Computer Services, Xerox), communications services vendors (e.g., AT&T, Bell Laboratories) and major system users (e.g., NASA, the National Communication Systems, the Air Force). The DISY group continues to meet, and to provide input to ISO/TC97/SG16.

Historically, the typical ISO standardization effort has lasted about ten years, from the standard's initial proposal to formal adoption. For distributed systems, a goal of three years has been set.

The international telecommunications standardization body, CCITT, has also shown interest in developing standards in this area, presumably so that more sophisticated "phone" service may be offered. CCITT also seems sensitive to the urgency in this area, and has been advancing the standards efforts relatively rapidly.

Where these efforts will go ultimately will only be determined in time. Standardization in this effort is viewed as a most important problem by all parties currently involved in the effort.

DIGITAL EQUIPMENT CORPORATION'S DECNET – A VENDOR EXAMPLE

In the Spring of 1975 Digital Equipment Corporation announced its intention to develop a computer system network architecture, DECnet. Since that time two phases of system products including DECnet have been announced. We will examine here briefly

1. Why DIGITAL decided to develop DECnet
2. What DECnet is
3. What DECnet capabilities DIGITAL's various products currently (Spring 1978) provide
4. How DECnet relates to existing and future standards

Why DIGITAL Decided to Develop DECnet

From the beginning, DIGITAL's minicomputers have been used for interactive applications. Also, many DIGITAL minicomputers have been used for communications applications, such as message switching and communications concentration for large timesharing systems. In the early 1970s the trend toward greater use of data communications became evident, especially as minicomputers received more and more use for data processing applications.

Although many computer networks had been built before DECnet was designed, there were few standards for digital communication that seemed germane to the wide range of applications for which data communications was likely to be used. Without standards of one form or another, digital computer networks seemed an unlikely eventuality. DIGITAL reasoned that if it were easier to build computer networks, more small computers would be used, since they could substitute nicely for larger machines in many applications. If more small computers were bought, DIGITAL – the minicomputer market leader – would benefit as well. Hence, DECnet was born.

DECnet was to be both a system architecture, and a set of products based on that architecture. The key concepts behind DECnet were these:

1. **Data packet oriented.** Following the successful ARPANET design, done under DOD sponsorship, the network would be based on the concept of data packets, which has been described earlier in this chapter.

2. **General purpose.** DECnet was to be a useful tool for most computer-to-computer interconnection problems. For example, DECnet should be usable to build a laboratory configuration, with one large central computer, and many smaller interconnected satellite computers that monitored equipment.

3. **Modular.** DECnet should divide the communication software into a number of functional layers or modules. Some applications would use the full DECnet capabilities; some might just use part.

4. **Practical.** DECnet should reflect the reality of minicomputer use.

   - DECnet software should be usable in compact subsets, so that a wide range of minicomputers could be interconnected, not just the biggest and most powerful.

   - DECnet networks should be buildable without requiring new communications hardware – existing communications equipment, particularly modems, should be usable.

5. **Non-proprietary.** DECnet would not be a proprietary design. Not only could the DECnet design be used by anyone without license or payment of any kind, DIGITAL would disclose the full details of existing and planned architecture standards and encourage their implementation by others.

The DECnet Architecture

The DECnet architecture is a formal specification for a data communications system. It includes definitions of a set of interfaces and, for the most complex interfaces, a set of protocols (detailed rules for interacting across the interface).

The intent of having these formal rules is twofold:

1. A program (either a user's application program or a system utility) can be written to interface to DECnet software, and can subsequently be used on any system that supports DECnet. In other words, having well-defined interface standards means that there can be program compatible interfaces to DECnet in many systems.

2. Systems written to these standards can communicate with each other across communications facilities (one of the standard interfaces).

As all who have written communications software have learned, these interface standards are a lot more detailed than one might suspect at the onset. The standards have to describe the conventions to be applied *in all cases*, not just the normal case. Unless detailed standards exist, certain problems develop.

- Program behavior will differ from one system to the next; application behavior will be unpredictable.

- The communication system will be unreliable, inefficient, or both, if it works at all.

The DECnet designers chose initially to specify three standard interface/protocol layers in the architecture.

1. **DDCMP.** DIGITAL Data Communications Message Protocol is the set of rules for using the communication facility, such as a phone line. DDCMP serves the function of a logical link manager (level 2), in the communication model introduced earlier. DDCMP is a packet-oriented, multi-drop line protocol. It performs error-detection and correction by transmission. The protocol is designed to be usable with existing communication equipment.

2. **NSP.** Network Services Protocol is the set of rules for building logical channels, initiating communications sessions, routing data packets onto logical links, etc. NSP covers parts of levels 3 and 4 in the model.

3. **DAP.** Data Access Protocol is a global file access mechanism that permits remote files to be opened and the data read and written over a DECnet network. DAP is an example of a remote file access utility, as discussed earlier, generalized to provide standard file access interfaces for all programs in the network.

Protocol Emulators. DIGITAL assumed from the beginning that DECnet would ultimately be only one of many schemes, and has made provisions from the beginning for interfacing to networks with other designs. In this case, DIGITAL software will emulate the "foreign" network protocols, making the DIGITAL computer act like, say, an intelligent terminal controller in the foreign network.

DECnet Products

DIGITAL has written software that incorporates a subset of the full DECnet functionality in all of the major software products, and has designed hardware to make communications more efficient. As of this writing, DIGITAL's software supports point-to-point data links (i.e., data can not be routed through an intermediary node).

Between end-points, the following program facilities have been provided:

- **Task-to-task communications** – The ability of a program running in one computer to send messages to and receive messages from a program running in another computer.

- **File transfer** – The ability to move files of data from one system to another over the network.

- **Resource access** – The ability of a program to use system facilities on another system, such as reading and writing files on that system, printing data, loading tasks and causing them to execute.

The following software systems implement (as system options) all three capabilities (task-to-task communications, file transfer, and resource access):

- PDP-11 software systems
  - RT-11
  - RSX-11S
  - RSX-11M
  - RSX-11D
  - IAS

- VAX-11 software systems
  - VAX/VMS

RSTS/E's (PDP-11) DECnet option offers task-to-task communication and file transfer capabilities. TOPS-10 (DECSystem-10) and RTS-8 (PDP-8) implement only task-to-task communications. If two systems are linked they may jointly use those functions common to both (for example, if a RSTS/E system is linked with a VAX/VMS system, then resource sharing may not be used on either system).

DIGITAL has also provided hardware options for use with DECnet. The DMC-11 is a specialized microprocessor that works with

PDP-11 systems to relieve the central processor of most of the processing necessary for DDCMP line protocol. Although DDCMP does not require the use of special hardware, the DMC-11 runs DDCMP communication at rates up to 1 megabit per second, over local, coaxial cable links. DIGITAL has also produced terminals (the VT-62) that utilize the DDCMP protocols for communication.

DECnet and the Future

With the exception of DECnet, and other vendor-provided standards, there are still no standards for general computer interconnection (CCITT X.25 is only a part of such a standard). Therefore, the reasons for creating DECnet in the first place – the absence of other standards – still hold true. DIGITAL is keeping close watch over both formal standards efforts, and the possible informal creation of de facto industry standards. As these standards develop, DIGITAL intends to support them, compatible with the best interests of DIGITAL customers. Until such standards eliminate the need for DECnet, it will be actively developed and supported.

SUMMARY

Electronic data communications interconnects terminals and computers in a distributed system. The past 25 years has seen a rapid and continuing growth in this area, as the existing telephone system is adapted for effective digital data communications, and as new special purpose systems are developed.

Chapter 11
TECHNOLOGICAL FORCES

The next three chapters address some of the key distributed system design issues. The problems and their solutions are presented at a general level. This handbook is not intended to serve as an implementation "cookbook," but can point to important problem areas and to some known solution methods.

The next three chapters cover the following topics:

- The present chapter (**Technological Forces**) discusses computer component technology – the ultimate force behind the increasing use of distributed systems.

- Chapter 12 (**Distributed System Design Goals**) presents key technical design objectives for a distributed system.

- Chapter 13 (**Distributed System Design Techniques**) discusses techniques that are used to achieve the established goals.

BACKGROUND

Technological improvement is the key factor behind the ever expanding use of data processing equipment. Computer component technology is the state of advancement of the data storage and processing devices used in computer systems. This technology has improved dramatically since the first commercial computers were built less than three decades ago. Rapid technological evolution continues today, despite the maturity of the technology. In fact, semiconductor technology evolution appears to be accelerating!

Improvements in component technology are applied to products in two ways:

- Computers with a specific price are made more powerful (constant price, increased functionality).

- Computers with a specific functionality are made less expensive (constant functionality, decreased cost).

Both approaches have been followed. *Increased functionality* in computers leads to new applications that were previously *too complex* for available computer systems. *Decreased cost* in computers *increases the market* for existing applications, and suggests new applications that were previously *too costly* using available computer systems. In each case, the component improvements have lead directly to increased use of computers.

INTEGRATED CIRCUIT (IC) TECHNOLOGY

The miracle technology is semiconductor integrated circuits (ICs). An IC is a complete electronic circuit fabricated on a fraction of a square inch of silicon. ICs are expensive to design and require costly manufacturing facilities, but they are very inexpensive to produce in volume. Where the volumes of production are high, IC technology provides complex electronic circuitry at very low cost.

The important points about ICs are:

- Initial design costs are high. Design changes are also very costly.

- Manufacturing requires expensive equipment and carefully controlled processing steps.

- At any point in time, there is a limit to the complexity of an IC that can be produced in volume. Beyond this limit, the production yield – the percentage of fabricated ICs that function as designed – falls off rapidly.

- Within the bounds of producability, the complexity of a circuit has little to do with its cost.

- *The maximum complexity of circuits that can be produced has approximately doubled each year since ICs first appeared commercially, in the early 1960s!* The cost-effectiveness of integrated circuits has roughly doubled each year[1].

With current IC technology, few circuits that require high-volume production are as complex as the technology permits. Instead, the highest production volume is for general-purpose building block circuits, with many different applications. Two key exceptions to this rule are memory circuitry and micro(scopic) processors.

IC Memory Components

Integrated circuitry is the most cost-effective technology for modern computer central memory systems; it surpassed magnetic core technology several years ago. With an IC technology central memory system, most of the circuitry in a computer system is dedicated to memory, providing a very large market for these components.

Continuing advances in IC technology have dramatically reduced the cost of computer central memory systems and have encouraged the use of bigger memories. Historically, a large amount of software design and programming effort went into program packaging in order to overcome the limitations of small central memories. As memory costs diminish, programmers concentrate more on creating useful program functions, and less on packaging.

---

[1]At any point in time, IC technology limits the complexity of practical IC devices. For digital logic, the complexity of an IC is the number of logic gates the circuit uses. The practical complexity limit roughly doubles each year. But once a specific circuit can be manufactured, the cost of the circuit does not decrease dramatically with further advances in technology. For practical circuits, the key cost determinant is the production volume.

IC Microcomputers

Today's IC technology permits an entire small processor and memory to be fabricated as a single circuit. Some of these IC microcomputers are used to build small computer systems. But most are used to replace conventional logic circuits inside electronic equipment.

For example, in modern computer terminals a microcomputer replaces many of the circuits used in earlier designs. Improvements in IC technology result in more powerful microcomputers that can be used to replace logic circuits in an ever-growing class of applications. IC microcomputers are already used in household appliances such as microwave ovens, and are used to control the engine ignition in some automobiles.

ICs and Computer Processors

Advances in IC technology make modern computer processors remarkably powerful by historical standards. The most advanced single-chip IC microcomputers are comparable to full-sized mainframe processors of the early 1960s. Today's minicomputers are comparable in sophistication to today's mainframes. Processing power is becoming less of a bottleneck for most data processing applications; data storage and communications are emerging as the prime performance concerns.

MAGNETIC DISK TECHNOLOGY

Another key technology is magnetic disk storage. Unlike simple and reliable digital logic integrated circuits, disk drives require moving mechanical parts, precision bearings, complex air filtration systems, and intricate linear electronics. The difficulties in construction notwithstanding, modern disk drives are an essential part of distributed computer systems because they provide large volumes of on-line data storage.

Disk technology (as measured by the cost per data-byte stored) has improved at a rate of about 35 percent per year. This rate of improvement would seem spectacular except that it suffers by comparison to the more rapid rate of IC development.

Disk drive reliability has also improved. Many modern disk drives exhibit a mean time between failures (MTBF) of more than 6,000 hours (nearly one year of continuous service), which is quite amazing considering the complexity and precision of the device.

The base technology in a disk drive is magnetic recording, which is measured by the number of bits that can be recorded on a unit area (e.g. a square inch) of storage medium. Over time, the recording density has been increased by improvements in recording head design, by advances in storage media, and by improved electromechanical design.

Disk drive performance has also improved because of the advances in IC technology. As complex electronics become less expensive, more electronics are used.

- Complex recording schemes and error correction schemes increase the effective data recording density.

- Complex electronics are used to replace mechanics; device performance and reliability are thereby increased. For example, in early disk drives the recording heads were positioned by complex mechanisms. In modern drives, the positioning is done with complex, electronic feedback (servo) circuits.

- Device control and management functions are placed in the drive to reduce central processor overhead.

- Diagnostic tests are incorporated in the drive so that drive servicing can occur without the use of the computer system.

- IC memory devices are used as a data buffer; the buffer improves disk subsystem performance by keeping often used data in rapidly accessible IC memory.

ELECTRONIC DATA COMMUNICATIONS TECHNOLOGY

A third important technology is electronic data communications. Unlike ICs and magnetic recording, communications technology is relatively mature: the desire to transmit digital data hasn't led to dramatic improvements in cables or radio transmitters.

However, the availability of inexpensive IC memory and logic circuits, and powerful, low-cost computer systems, has led to major improvements in digital data communications.

Improved Use of Communications Links

Communications links are managed by dedicated computer systems. The sophisticated, computer-managed communications protocols permit higher data transmission rates without transmission errors. Computer management also permits many independent data transmissions to be multiplexed onto one communication link to increase the utilization of the link and decrease the cost of service.

Minimized Need for Data Communications

In distributed system designs, processors and memory can be used instead of communications for some functions. Instead of repeatedly retransmitting commonly used data to a terminal, we add a processor and memory to the terminal (create an "intelligent" terminal), and program the terminal to remember recently used data. Additionally, some processing functions, such as simple input editing, are performed by the intelligent terminal to further reduce the need for data communications.

SUMMARY

Computer system usage has increased due to the remarkable improvements in component technology, particularly IC technology and magnetic recording technology. In the last twenty years the improvements have been phenominal, and the improvements are continuing today. Differing rates of technological advance lead to a changing use of the technologies: as IC memory and logic become relatively less expensive, more ways are found to use ICs.

Chapter 12
DISTRIBUTED SYSTEM DESIGN GOALS

This chapter introduces key design goals for effective distributed, interactive data processing systems. By the time system designers and implementers establish specific design goals, we assume that the following has already occurred:

1. Top management recognizes the potential of distributed systems, as well as the major pitfalls. They have established general goals for the organization, including decision-making policies, and policies on openness and access to data.

2. EDP management, following the direction set by general management, has established an overall development direction, including, for instance, choice of major vendors, style of equipment (e.g., big centralized systems, small systems, or both), and policies for system development and management.

3. Departmental management is aware of the potential in distributed systems and the general direction for system development within the enterprise. With this in mind, department managers have analyzed information needs and flow within each department and between departments, and they are actively involved in specifying automation for their own departments.

Within this framework of general understanding and planning, we proceed to detailed technical planning. Many of the goals for a system are particular to the intended end use. Other goals are universal and apply to other systems:

- **Availability.** Keeping the system usable.

- **Integrity.** Keeping the data accurate and consistent.
- **Responsiveness.** Not delaying the user.
- **Security.** Controlling access to data and programs.
- **Privacy.** Protecting individual's rights.
- **Adaptability**. Planning for change.

Each of these topics is discussed in more detail below. Methods for achieving these goals are discussed in the next chapter.

AVAILABILITY GOALS

The availability of computer systems is important. A system is unavailable if it cannot be used when needed. In the past, computer systems were incidental to a job. For example, an early order entry computer system processed batches of punched-card order data that had been transcribed from a paper form. The computer system could be down (broken) for hours or even days with a negligible effect on order processing.

The situation is very different with an on-line, interactive, order processing computer system. If an on-line system fails, many more individuals are immediately affected.

Lowering the failure rate increases the cost of the system. Consider the following when setting availability goals.

- Any system *will* fail.
- You need to understand the *full* impact of the failure on the enterprise.
- There are many different ways of dealing with system failures.
- Choose an approach that is cost-effective for the enterprise as a whole.

Evaluating the full impact of a system failure is difficult: most people don't like thinking about failure; some of the impact will be subtle but significant. Some of the failure costs will be obvious, such as lost productivity. Some costs are indirect but important, such as lost customer satisfaction.

System availability is normally specified by the Mean Time Between Failures (MTBF) and the Mean Time to Repair (MTTR). A careful analysis can relate MTBF and MTTR statistics to the failure cost. Setting availability goals for a system helps the software designer identify the level of effort needed in this area, and helps the system owner select the correct level of service support from hardware vendors (e.g., on-site service staff, or service as needed).

Distributed systems have potential availability benefits *and* liabilities. Having multiple computers in a distributed system may mean that if one computer fails, some part of its functions can be absorbed by another. On the other hand, a distributed system has many component parts, each of which may fail. Distributed systems should not be designed so that correct system functioning requires that *all the parts work all the time.*

DATA INTEGRITY GOALS

The value of data accuracy depends on the use of the data. Inaccurate or inconsistent data is often worse than no data at all. Some data, such as accounting data, must be as accurate as possible. Other data, such as planning estimates, are intrinsically inaccurate.

There are many ways in which data may become incorrect:

- **Poor specification.** The definition and intended use of each data item must be clearly specified. Otherwise the person supplying the data may misinterpret the definition and provide incorrect data.

- **Entry errors.** Typographical errors may occur when the data is entered.

- **Programming errors.** An incorrect program may process correct input data but produce erroneous results.
- **System software failure.** An operating system failure may alter data by accident.
- **Hardware failure.** An equipment failure may damage data.
- **Media failure.** A defective or damaged storage device, such as a disk pack or tape reel, may result in the loss of data.
- **Operational failure.** The system operator may make a mistake, such as loading the wrong tape reel or turning off power at the wrong time.
- **Faulty interpretation.** Correct data may be incorrectly interpreted by the user.

Increasing data integrity increases the cost of the system. When setting data integrity goals, consider the following points.

- Some data will be damaged.
- Data differs in value.
- There are different ways of dealing with the failure; most systems use more than one method.

The value of data to the enterprise must be understood before effective system design can be done.

The value of data integrity increases in a distributed system.

- More individuals depend on system operation to do their jobs.
- Data is more valuable because of wider use.
- Data is used at a distance; for many users errors in data will not be obvious.

SYSTEM RESPONSIVENESS GOALS

The speed with which the computer system responds to a request contributes directly to the productivity of the system users. Poor responsiveness increases the cost to *use* the system. Adding more equipment to improve responsiveness increases systems costs. An ideal design maximizes total system cost-effectiveness.

Responsiveness issues permeate all of the system design. Achieving adequate responsiveness depends on understanding the value of responsiveness for each system function.

- Rapid response is most important when the system user has nothing to do until a response arrives. It may be more cost-effective to restructure the work flow than to improve the system responsiveness.

- Response predictability is as important as average response delay. An individual will learn to live with predictable response delay.

- In general, response delay should correspond to the user-perceived complexity of the request: the response to trivial requests should be rapid. Response time may be defined as the delay period after which 95 percent of the responses will have begun. In this measure, the following interactive request response goals are typical.

    1. **Input editing: .7 seconds.** For example, verifying that a check-digit is correct is editing.

    2. **Data query: 3 seconds.** An example is checking the quantity of a part in inventory.

    3. **Data update: 7 seconds.** An example is placing an order to be filled from inventory.

- Pursuit of many *other* system goals will degrade system responsiveness. For example, many mechanisms for improving data integrity will degrade system response.

Achieving adquate system response in a distributed system is a major design challenge.

- There are many system mechanisms that can limit system response, such as data access or data communications.

- Designs that are suitable for single computer systems cannot always be extended to multiple system designs because of the delays introduced by communications links.

The most important design step is to partition the entire system carefully into smaller, more manageable subsystems.

SECURITY GOALS

Unlike the goals discussed thus far, security is a paradoxical goal. Given an investment in computer hardware, and a value for user productivity, an optimally responsive system can be designed. *But the more secure a system is, the less potentially usable it is!*

A typical information system performs two distinct functions.

1. The system is a clerical tool used for gathering, storing and disseminating the operational data of an enterprise.

2. The system data base provides an invaluable management and planning resource.

Adding security to the first function is relatively straightforward. It is the second function that causes problems.

The general security problem is this:

- Data has value to an enterprise.

- The enterprise incurs a cost if some data is made available to individuals outside the enterprise.

- The enterprise incurs a cost if some data is changed or deleted.

- Each system is vulnerable to some level of security attack.

- Improving security defenses increases the system cost.

There are both external and internal security threats. Industrial espionage exists in many industries. Banks and other institutions are vulnerable to criminal actions. But the most important threat for many enterprises is the potential impact of a disgruntled employee.

The general approach to improving system security is limiting the data and functions available to each specific individual who is permitted to use the system. This approach works quite well with operational data processing because reasonable actions can be specified for each individual in the process.

Making management information functions secure is much more difficult. System-provided information is valuable to all levels of management, particularly to the line managers who run daily operations. The information is most valuable if it can be accessed in a totally free manner. Systems with the most usability have the least security.

It is difficult to assess the value of management insight provided by access to system data. The system designer can get some help by understanding the management style of the enterprise. If the enterprise management strictly controls access to information in general, the focus should be on security, not flexible access. Conversely, if the enterprise management encourages information dissemination, the system design should focus on making data usable.

As systems become more distributed, new security challenges appear.

- The greater use of data communications increases areas of vulnerability. The designer must anticipate digital data wiretapping as well as improper terminal usage.

- Many common sense protection mechanisms lose their value at a distance. Terminal access to a local system is protected just as access to local file cabinets is protected. A stranger's business will be questioned. That level of protection disappears when a system is typically accessed from a remote location, especially if dial-up telephone lines are used.

The value of data in a distributed system is not always obvious. One sales record may be considered to have little or no security value; but access to many of those records at once, so that patterns are discerned, may have a large security value. In some cases the volume of data traffic may have as much security value as the data itself.

DATA PRIVACY GOALS

Privacy pertains to personal data such as medical records or criminal records. Most societies explicitly recognize the rights of each citizen to some level of privacy, which means that under many circumstances, access to personal data cannot be made without the explicit approval of the individual. With the advent of computers, laws concerning privacy have been extended and applied to computerized data. Designing systems that provide adequate protection to personal data is very difficult. *But, an individual's right to privacy is not reduced because privacy is difficult to achieve.*

Privacy is fundamentally different from general data security. Breaches of security have a bounded cost. The value of sales information is related to the volume of sales and markets. Privacy is quite different.

Violation of an individual's privacy is an act against them in the eyes of justice, and the wronged individual is entitled to compensation. The amount of compensation is not related to the value of the data *to the wrongdoer*. If you injure an individual with your automobile, the resulting compensation has no relationship to the damage done to the car.

Privacy laws will not be waived because privacy cannot be achieved in a cost-effective manner. In many cases the best approach will be to eliminate personal data from the system data base because the value of the data doesn't match the potential liability of its misuse.

SYSTEM ADAPTABILITY GOALS

The need for an information system to adapt to changes in technology and system usage has been discussed at length in previous chapters. Enterprise management must create an initial design framework by specifying the growth and evolution goals of the enterprise. EDP management must have a general technical development plan that makes the goals achievable.

But ultimately the burden rests on the system designer. The critical steps are:

1. Decomposing the system into smaller subsystems *such that probable growth and division occur at established boundaries.*

2. Specifying and using design and development standards that provide some level of data and program interchangability.

Distributed systems require more careful formal planning than centralized systems. Design and construction philosophies that worked when all the programmers and computers were at one location may not work with distributed systems.

SUMMARY

In addition to the functional goals that are unique to each system, some universal goals exist. The goals reflect the costs and benefits of using computerized information systems. In the case of each goal, moving from centralized to distributed systems brings new design challenges. Techniques for achieving these goals will be discussed in the next chapter.

Chapter 13
DISTRIBUTED SYSTEM DESIGN TECHNIQUES

A collection of design ideas is presented for each of the goals discussed in Chapter 12. The collection is not exhaustive, nor are the techniques presented in great detail. These techniques may suggest alternatives that can be pursued further by referencing other technical literature or by seeking expert advice.

Where possible, make use of these techniques by using existing, tested software. Few computer applications justify extensive software development. Using standard software products dramatically reduces software development costs and total system costs.

When selecting a software vendor, be sure that the vendor is able to respond to your problems now and in the future.

TECHNIQUES FOR IMPROVING SYSTEM AVAILABILITY

Any piece of computing equipment will fail; the question is what to do until it is fixed. Alternatives range from on-board satellite computers that fix themselves, but are very expensive, to systems that can be broken for extended periods without causing a major disruption. Some intermediate alternatives are discussed below.

A Hot Standby System

A hot standby system is a complete spare computer system, which is powered up and loaded with the application software. If the application system fails, load is transferred to the standby system. A variant of the hot standby is called "N+1" redundancy: each critical component has at least one available spare – if 4 disk drives are needed the configuration has at least 5. "N+1" systems are typically less expensive than fully duplicated systems.

A Cold Standby System

A cold standby system is a separate computer system that can replace the application system in the case of failure. In constrast to a hot standby system, it performs other tasks and must be initialized before it can assume the application. Typically the back-up system is used for program development and batch processing.

Load Sharing

With a load-sharing approach, when an application system fails, the processing load is moved to another system. The back-up system may be located at the same site as the failed system, or it may be at a remote site and accessed via communication links. The back-up system absorbs the additional processing *and* continues to perform its primary tasks. Under the combined load, service may be degraded. The designer may choose to move only the most critical functions to the back-up system.

Alternate System

An alternate system can be used for some or all of the functions provided by application system, in the case of failure. For example, an on-line minicomputer is backed up by a central mainframe system or timesharing service.

Paper Back-up System

With a paper back-up system, some of the application functions are continued without any computer. During normal system operation a set of paper files is printed periodically. In the case of failure, use of the documents replaces use of the on-line computer. For example, a branch bank backs up some on-line system functions with account status reports printed nightly.

Failure Detection

The techniques previously discussed are used after a system has failed. The problem remains of detecting system failure. Quick and total system failure is easy to detect, but occurs only rarely. A more insidious type of failure occurs periodically. On-line software systems have many internal consistency and "sanity" checks, designed to detect failure soon after it occurs, so that remedial action can be taken.

TECHNIQUES FOR INCREASING DATA INTEGRITY

There is no simple way to insure data integrity, because virtually any kind of failure or error can alter on-line data. A careful system design can minimize the impact of these accidents. For each of the sources of integrity problems listed previously, techniques are given below.

Data Errors Due to Poor Specification and Faulty Interpretation

Data specifications should be clear and easy to understand. Accurate and up-to-date documentation must exist for data files, and for the programs, reports, and data entry forms that use those files. The user documentation must be well-written, and must accurately reflect the data definitions.

Data Entry Errors

Data entry errors can be minimized by:

- careful operator training
- careful human-factors engineering of all user interfaces (for example, careful selection of terminology)
- easy-to-use data entry forms
- check-digits and other consistency-testing mechanisms

- on-line editing of input data, with immediate error diagnostics
- internal consistency checks that compare input data to the on-line data base (For example, is an order of this size reasonable for the customer?).

The best time to detect and correct input errors is when the data is being entered. At entry-time, the other information at hand simplifies the correction of errors.

Programming Errors

Designing and testing programs is an entire subject by itself, but the key is creating thorough program specifications *before* the program is written.

System Software Failure

Choose a vendor with a reputation for high-quality software and after-sales service.

Hardware Failure

As with software, choose the vendor carefully. Effective hardware maintenance requires an established service staff and an inventory of spare parts. The best guarantee of service is a nearby staff and a local inventory of replacement parts.

Storage Media Failure

Some media failure is inevitable. Some failures are automatically handled by the storage subsystem (for example, a temporary disk pack read error); others require that a new copy of the data be created from a back-up version. The frequency of failures that affect system operation depends on

- the quality of the media used
- the quality of subsystem maintenance
- handling care

Operational Failure

The two steps to minimizing operational errors are

1. Replace manual procedures with automatic procedures where possible.

2. Carefully document the remaining manual procedures, and train the operators.

Many manual procedures can be replaced by an automatic procedure. The automatic procedure can often be initiated automatically, on a prespecifed schedule (e.g., perform end-of-week accounting every Friday at 9 PM). Where manual control is necessary, a complex procedure can be packaged into a single, simple system command. For example, the single command "BACKUP" can replace a long sequence of file transfer commands.

Data Base Redundancy

As we use the term here, a data base is redundant if erroneous data can be detected by inspection. For example, with some additional tables, we could detect when a street and city address do not agree with a postal ZIP Code (the ZIP Code is redundant data; in the case of an error, the address and ZIP data are inconsistent). The classical, double-entry bookkeeping scheme has built-in redundancy in the data – the accounts will not balance in the case of most errors.

The redundant data used for consistency checking increases the storage requirements of the application. Consistency checking is additional processing. These extra costs are to be incurred only where justified by the benefit returned.

These are some common consistency techniques:

- In batch processing, a new master file is checked for consistency before the file update job is deemed to have succeeded.

- Utility programs that check for suspected data damage are written. For example, the utility program is run after an application program fails to complete properly. The utility program can be designed to "garbage collect" the data base, which means doing as little as is necessary to make the data base consistent again.

- In on-line systems, consistency-checking utility programs are run periodically to validate the data base consistency. The background program initiates emergency recovery actions if a failure is detected.

- Application programs are designed to validate data base consistency during normal processing (e.g., Are the values in this accounting data record within established limits?).

The goal of these consistency techniques is to identify errors quickly so the data base can be repaired before the erroneous data is used.

Data Base Recovery

Recovering a data base means restoring damaged data to a valid state by using copies made before the damaging accident. Recovery uses two sources of input:

1. **Back-up copies.** Periodically, parts of the data base are copied so that an alternate source of the data will exist. The data base may be copied on a physical volume basis, which means that an entire storage volume (e.g., a disk pack) is copied at once; or it may be copied on a more specific basis, which means that only specific data collections are copied.

2. **Modification journals.** A journal is a data file that records updates to the data base. When a record is updated, the previous value (the "before image") is recorded in the journal along with the new value (the "after image"). The journal includes time-stamp records that give the time of each record update, and synchronization records ("quiet points") that indicate when the data base was consistent. For example, if an

application program updates six data records as the result of a single transaction, a single synchronization point will be written when the entire transaction is finished.

Data base recovery consists of using back-up copies and the modification journal to restore the data base to a consistent state after damage has been discovered. The goal of recovery techniques is to include all updates that occurred until the time of the accident, in order to lose as little work as possible. Recovery can be either done backward, moving from the current state incrementally backward in time, or recovery can be done forward, moving incrementally forward from a back-up point.

Backward Recovery. The data base has been damaged. System operation is temporarily suspended, and recovery begins. Backward recovery consists of these steps:

1. Read the modification journal file backwards, one update record at a time. The records are read in the inverse sequence from the order in which the updates occurred. The first record read is the last update that occurred before the system failed.

2. For each record read, use the before image record from the update record to restore the data base to the state that existed before that update.

3. Continue Steps 1 and 2 until reaching the first quiet point that occurred before the failure.

Forward Recovery. Instead of logically moving the data base back in time, forward recovery begins with a back-up copy and uses the modification journal to advance it in time.

1. Position the modification journal to the point in time at which the backup copy was created (by using the time-stamp records).

2. Read the modification journal in the forward direction, one update record at a time.

3. For each record read, use the after-image to update the data base.

4. Continue Steps 2 and 3 until reaching the last quiet-point before the damage occurred.

Commitment-based Recovery. The recovery techniques previously discussed restore the data base to a consistent state after an accident. Most data processing systems include such mechanisms as insurance against major problems: a major subsystem failure, loss of power, or damage to a disk pack.

System operation must be suspended during forward and backward recovery. While this is reasonable for major failures, there are many transient or localized failures that are better handled by commitment-based recovery.

Commitment-based recovery is designed to handle failures that are discovered between the beginning and end of a single transaction, where a transaction is a set of data base updates that must be completed as a unit in order to keep the data base consistent. For example, a sales update transaction may update many different data records. If a failure results in some but not all updates occurring, the data base will be inconsistent.

Commitment-based recovery consists of deferring the actual data base update until the application program signals commitment to the entire transaction. A data base update consists of:

1. Reading part of the data base from disk storage into central memory.

2. Updating the data base in central memory.

3. Writing the updated copy back out to disk.

Deferring the update means deferring Step 3 until the commitment signal is received. If a failure occurs before the signal is received, the transaction processing is restarted and the previous work is forgotten.

Multiple System Recovery

All recovery schemes require synchronization of processing activity and data base modifications. The easy part of recovery is replacing damaged data records with clean copies; the hard part is synchronizing related updates, and keeping the data base consistent.

The schemes discussed can be extended to distributed systems provided the systems can synchronize their activity. Careful attention must be paid to multiple system designs because of the delays introduced in data communications. The performance impact of commitment-based recovery depends on the time period for which updates are deferred: the shorter the better. Adding a second or two to these delays, in order to synchronize with remote systems can have a large impact on performance.

TECHNIQUES FOR IMPROVING SYSTEM RESPONSIVENESS

Achieving system response goals is a general problem. The key factor is decomposing the system into localized subsystems. This has been discussed at length already. In this section we will examine the problems of high-performance data storage, because data access (not processing) is becoming the primary system performance constraint. The discussion will cover these topics:

1. The disk "problem"

2. Allocating disk storage to maximize performance

3. Data caching for further performance improvement

The Disk "Problem"

The cost-effectiveness of disk storage – the cost per data byte stored – has improved continuously since disk drives were first used commercially in about 1960. But the performance of disk drives, the average access time *per megabyte stored*, has gone down over time. As a result, disk management techniques have changed; the use of fast buffer memory is key to high disk performance.

A disk drive consists of two subsystems:

1. An electromechanical subsystem – a rotating motor to spin the disks and a linear motor to position the recording heads across the recording surfaces

2. An electromagnetic subsystem – the magnetic recording material used to coat the disk surfaces, the recording heads used to read and write the data, and various electronics used to connect the heads to the computer system

Over time the performance of the electromechanical subsystem has advanced relatively little, because of physical constraints such as inertial mass, and because disk drives are not a significant part of the total use of motors (the rapid growth in disk use has made little impact on the total use of motors for all applications). In contrast, the number of bits that could be recorded on a square inch of disk surface has gone up by about 35 percent per year, due to advances in the electromagnetic subsystem.

The imbalance between progress in electronics and mechanics leads to degraded disk performance. For example, at some point in the past there were disk drives with the following characteristics:

- cost: $25,000
- capacity: 25 million bytes (MB)
- average access delay: .040 seconds (40 ms.)

For a specific application we require a storage subsystem with 100 MB capacity. We use four disk drives, and the resulting subsystem has the following characteristics:

- cost/MB: $1,000
- capacity(MB)/access(ms.): 10

The second characteristic is called the access bandwidth. It measures the speed with which we can access the data, taking into consideration the size of the data store. If we double the capacity of the subsystem, and the average access delay remains the same, then the access bandwidth doubles: we can get to twice the data in the same period of time. If the capacity remains the same, but the access delay is halved, then the access bandwidth again doubles: we get to the data faster. We compute access bandwith because data volume and access delay are both critical factors in application capability.

Assuming that the four drives can be used independently, the average access time for the subsystem is 10 ms. (we can perform 4 times as many accesses; the average access time must be one-fourth), giving the access metric value of 10, shown on the previous page.

About 5 years later, disk drive storage costs have gone down by a factor of 4:

- cost: $25,000
- capacity: 100 MB
- access delay: 40 ms.

At this point we can build the 100 MB subsystem with a single drive:

- cost/MB: $250
- capacity(MB)/access(ms.): 2.5

The storage cost has gone down by a factor of 4, but so has the subsystem performance. The effect is caused by the use of mechanical devices in a disk: the disk unit has not been shrunk, we've just used advances in electronics and magnetics to store four times as much data. In striking contrast, when the price of a computer system is reduced by a factor of 4, due to advances in electronics, it has the same performance.

Effective systems must have balanced processing and I/O capacities. In order to build high-performance, low-cost, computer systems, we must find techniques to improve disk performance. There are two classes of techniques:

- Optimally allocating the location of data on the disk surfaces

- Buffering frequently used data in faster storage (caching)

Disk Allocation

The major components of disk access delays (disk latency) are

- **The time required to move the heads across the disk surface.** This varies from no delay if the heads are positioned on the correct data cylinder; about 10 ms. if the desired cylinder is adjacent to the present cylinder; up to about 40 ms. to move across the entire disk surface.

- **The time required to rotate a data sector under the disk heads.** A typical disk spins at 3600 revolutions per minute; a rotation takes 17 ms. On average, it will take one-half a rotation for the data, or about 8 ms.

Disk performance can be substantially improved by placing the data on the disk surfaces so as to minimize the average access latency.

- The most frequently accessed data should be clustered together into a small area. With an optimally allocated disk surface, the frequency of access is highest for the center cylinders, and then falls off for the outer and inner cylinders.

- Data that is used sequentially should be stored on the disk surface sequentially, and should be read and written in large contiguous pieces, not in individual blocks. Once the disk is at the data, large volumes of data can be transferred to memory quickly. Moving the heads and waiting for rotation is what hurts performance.

Data allocation can be done when the application is designed, or the disk can be periodically restructured, on the basis of observed access frequency. Even when the allocation is optimal, further performance improvement can be gained by the use of data caching techniques.

Data Caching

Data caching is keeping a smaller buffer of data near where you need it. For example, data that is permanently resident on a disk pack may be temporarily cached in central computer memory where access is much quicker (1,000–10,000 times quicker, typically).

The percentage of time needed data is found in the cache, and we do not have to use the disk, is called the hit-ratio (HR). The average access time of the entire storage subsystem (cache and disk) is computed as

$$HR * access_{cache} + (1-HR) * access_{disk}$$

Where the access time varies widely between cache and disk, especially when the disk is on a remote system accessed via a communications system, a cache can dramatically improve the average subsystem access time.

The faster a data storage device, the more expensive it is. Unless the cache buffer is small compared to the disk, the cost of the cache will dominate the cost of the subsystem. But if the cache is relatively small, the choice of which data to cache must be made intelligently, or the hit-ratio will also be very small.

Caches are effective because data usage is statistically predictable with simple algorithms. We can rarely predict which specific data item will be used next, but we can keep a small collection of data items with a high probability that the next access will come from that collection.

Two independent effects make data caching effective:

- **Data reuse.** A data item that has just been accessed is likely to be accessed again soon.

- **Sequential access.** A data item is more likely to be accessed if the item logically preceding it has just been accessed.

Cache Applications. Modern operating systems use portions of central memory as a selective disk cache to improve effective disk subsystem performance:

- *File directory structures.* The file directory relates account and file names to the disk areas used to store the file data. Keeping parts of the directory data in memory reduces the time needed to open files.

- *File access windows.* A file access window maps the logical blocks of a data file into the physical storage blocks containing the data. The operating system uses the access windows to translate logical file addresses used by an application program into physical disk addresses. Keeping some of these maps in main memory reduces the number of physical disk accesses needed to complete a logical file access, particularly for sequential file accessing.

- *Record key index blocks.* Index blocks are used for rapid access to data records on the basis of a specified record key. Caching the root index blocks in main memory (the ones that are used for most accesses) substantially improves the average access performance by minimizing the physical disk accesses needed.

- *Anticipated data blocks.* The operating system tries to recognize sequential file access. If an application program reads file-block$_N$, and next reads file-block$_{N+1}$, the file system automatically reads and buffers file-block$_{N+2}$, (file-block$_{N+3}$, ...). These blocks are probably stored on the disk near the requested block, and can be moved to main memory in very little additional time.

- *Most recently used data blocks.* The operating system uses a part of main memory as a disk block pool. All disk reads and writes are from this pool. When a read is requested, and the block is not already in the pool, and there are no unused buffers in the pool, then the buffer that has been used least recently is released.

Caches in Distributed Systems. Distributed systems have the same data access problems, but access times are lengthened because the disk storage unit is located at a remote site and accessed via communication lines. Since the average access to the disk is longer, the impact of a cache is greater.

The use of local caches also minimizes the use of data communications. Unlike the case of access to a local disk, we must pay for data communications, often based on the volume of data transmitted.

The following are a few examples of distributed system caches:

- *Intelligent terminals.* An intelligent terminal is a terminal and a small computer. The computer can be programmed to perform data caching functions. For example, if the terminal is used for form-based data entry, the most commonly used data forms can be cached in the terminal.

- *Front-end communications processors and concentrators.* A front-end processor moves communications processing nearer to the terminals. A front-end processor is like a timesharing intelligent terminal.

- *Data caches.* Distributed systems are designed so that data used in one locality for a period of time is temporarily moved to a system near the use.

Cache Design. Caches work because local copies of data are made. Making copies of data is a major source of problems. For example, if two programs each have a copy of a specific data record, and each updates the record at about the same time, then one update will be lost (whichever one is stored on the disk *first).*

The obvious solution approach is to restrict simultaneous access to data. Depending on how this is done, the effect on performance can be disastrous. If only one program can access the entire distributed data base at a given time simultaneous update problems will be avoided, but the performance will be terrible. On the other hand, controlling access to a single record requires more complex management.

Cache design also must anticipate the possibility of equipment failure. In a distributed system, the communication network may fail after data has been cached in a remote system. This kind of failure must be addressed in distributed system cache design.

TECHNIQUES FOR IMPROVING DATA SECURITY

The keys to data security are

- a careful analysis of the informational needs of an enterprise
- thoughtful, ongoing management of the information system.

A careful analysis of needs is the only means by which useful security controls can be applied without seriously degrading the usability of the system. Security needs change over time, with changes in the system and system usage. To be effective, system security must be managed as an ongoing effort.

Logical Security

Security is implemented by logical partitioning.

1. Divide the data base into security categories. Some data records are sensitive by themselves, others represent sensitive data when aggregated.

2. Similarly, categorize the system procedures or transactions. Some transactions are sensitive, some transactions are only sensitive with certain data.

3. Categorize means of system access. In general, access via a dial-up telephone line is more dangerous than access from a terminal in the computer room.

4. Finally, assign system privileges in the form

$$\text{permit}\ (\text{user}_W, \text{transaction}_X, \text{data}_Y, \text{location}_Z)$$

$User_W$ is permitted to execute $transaction_X$ on $data_Y$ by using a terminal at $location_Z$. In most applications the privileges can be grossly simplified; each user is given access to certain transactions, for example.

Physical Security

The value of physical security should not be underestimated. A modestly designed computer system is no more or less secure than file cabinets in the same location. One of the advantages of small, local computer systems that are not accessed from remote sites is physical security. Even if there is little logical protection in the system, a penetrator must intrude to gain access to the system. An intruder could rifle disks or file cabinets as well.

Communications Security

Security is a greater problem when a system can be accessed remotely. Then the penetrator is not necessarily an intruder. Grand larceny can be committed from the comfort of your home, by telephone. Any system that permits remote access should have some form of user identification and password exchange. Such checks will at least eliminate the casual wrongdoer. Beyond simple password protection, these techniques are available:

- *Leased communications lines.* The link to the system is prewired, instead of being formed by a dial-up telephone connection. A leased line is expensive but adds a level of protection by making it more difficult to mimic a valid user.

- *Dial back.* Requests for system service are made via a conventional dial-up phone connection, but service is performed only after the computer system phones the user back, at a prevalidated phone number. This is a poor man's leased telephone line. The system can be used only from valid phone numbers.

- *Encryption.* Modern semiconductor technology has led to inexpensive data encryption equipment. An encryption device takes an input data stream and produces a coded output data stream. The particular code used is specified by an encryption key that may be changed whenever desired. Encryption has two benefits.

  1. The encrypted data is practically impossible to understand, without preknowledge of the encrypting key.

  2. If the encrypting keys are carefully protected, receiving a message encrypted with the correct key is further confirmation of the sender's identity.

Continuing advances in IC technology will make encryption an inexpensive communications option. Encryption does not mean perfect security, but it does reduce the potential of security threats due to simple wiretapping.

Audit Trail

A good security system quickly identifies when a security problem exists. The basic method is keeping a detailed security audit trail – a data file that records the occurrence of any significant event. The log includes records indicating when users logged into and out of the system, when sensitive transactions were executed, and when sensitive data was accessed. In most cases the security audit trail is insurance, like the photographs taken automatically in a bank but not developed except when needed. But if there is a question of a security problem, an audit trail provides valuable data.

TECHNIQUES FOR BUILDING ADAPTABLE SYSTEMS

Dealing with change and evolution has been discussed at length already. Good programming practices are essential to building adaptable systems. There are good texts on software engineering practices, and we will not go into the subject here.

Our focus here is on the use of standards. Many different individuals will design, build, modify, maintain, and use complex, distributed information systems. Standards are one important means of coordinating independent work.

Standards are particularly significant for distributed systems that communicate with each other and perform cooperative processing. *Without interconnection standards, there is no communication and no cooperation.*

Standards don't represent the *right way* to do something; they represent the *economical way* to do something. Standards should be chosen in order to maximize savings, and not on aesthetic grounds.

- Use of standards maximizes the reuse potential for software and systems.

- Use of standards minimizes programmer training costs.

- Use of standards provides for the broadest choice among vendors and promotes a competitive marketplace.

Different levels of standardization are used for different reasons.

1. **Informal standards.** Standards don't have to be written in legal terminology and published in policy notebooks. Meaningful, informal agreements provide much of the benefit of formal standards, and are easier to develop and install.

2. **Formal standards.** As an organization grows or spreads geographically, informal standards must be formalized. Developing and installing formal standards is often a painful process. Every reasonable effort should be made to prepare complete, comprehensive, highly understandable, and far-sighted standards. Once a standard is installed, conformance to the standard must be monitored. If a standard proves inadequate, the standard should be modified or abolished, not ignored.

3. **Vendor-supplied standards.** Major computer system vendors develop and use internal standards. These standards can be adapted for other uses.

4. **National and international standards.** The United States has a national standardization body (the American National Standards Institute - ANSI), which acts as the national representative to the International Standardization Organization (ISO). ANSI standardization efforts are supported by computer vendors, government agencies, and major computer users. The resulting standards become requirements for many government computer contracts, and are often adopted voluntarily by manufacturers.

At this time, no national or international standards adequately define distributed system interfaces. Most major equipment vendors have defined networking standards, but these standards are incompatible, and in many cases incomplete and not fully developed. The standardization community recognizes that distributed systems require meaningful standards, produced in a timely fashion. General interconnection standards that apply to equipment produced by more than one vendor seem needed in the future, but are by no means guaranteed. For now, the computer user that expects to benefit from standards must develop those standards.

SUMMARY

A set of design techniques for general distributed system problems has been presented. The techniques are not cookbook solutions, but rather possible approaches to study with the use of additional material or consultation.

Chapter 14
A PEEK AT THE FUTURE

The major portion of the information used in offices today is still stored in the form of paper and is transmitted manually or mechanically. But the percentage of information managed electronically is growing. Electronic data storage, processing, and communication offers many advantages:

- lower cost
- greater accuracy
- greater accessibility
- faster delivery

Computer technology continues to improve at a rapid rate. Better technology leads to cost/performance improvements in computer equipment, which in turn lead to new and expanded use of computer equipment.

The future of computer systems cannot be predicted. It depends partly on technology and partly on the decisions made by major computer producers and consumers. In many cases, the potential offered by technology is realized only in high-volume products that require a large initial investment.

DATA TRANSMISSION

Digital data transmission is the key to future developments. In time, transmitting data efficiently will become as simple as using the telephone.

- The existing telephone system uses digital transmission techniques extensively.

- Rapid improvements in integrated circuit technology are forcing a changeover from electromechanical to electronic telephone switching equipment; the electronic equipment uses digital techniques internally.

- Communication satellites establish high-capacity transmission channels without investment in cable or microwave facilities.

OFFICE AUTOMATION

There is a critical-mass effect involved in the use of information systems. The systems are very attractive when they are seen to be in widespread use; it's hard to be a pioneer.

Today's technology is adequate to build much more elaborate information systems than are in common use; everyone is waiting for the pioneers. If system technology cost/performance continues to improve, ultimately the systems will appear. Word processing and other office automation technologies are good examples in point.

Word Processing

A word processor is an electronic typewriter. Its advantages over an electric or manual typewriter, as seen today, are ease of use and productivity, especially for complex, technical documentation where accuracy is important (for example, legal text).

The impact of word processors will come from the greatly increased percentage of information that will exist as digital data. When a key is struck on an electric typewriter, the mechanism imprints the character image on the page. When a key is struck on a word processor, the character code is stored in a digital memory. Today the word processor memory permits the document to be corrected and modified without the probability of additional errors being made during retyping. In the future, word processor-like terminals will send and receive documents via electronic communications networks.

Electronic Mail

Electronic mail is the sending and receiving of messages as digital data rather than as paper documents. Many specialized applications of electronic mail exist today; most large companies have an internal message switching and delivery system.

Electronic mail systems will ultimately transmit a much larger percentage of messages. Between now and then

- Word processing equipment, with communication capabilities, will become widely used.
- Data communication will become as simple as voice telephone communication.
- The usage critical-mass will be achieved.

The potential of electronic mail is great:

- **Timeliness.** Electronic delivery takes seconds or minutes.
- **Economy.** The labor and energy invested in electronic document preparation and transmission is less.
- **Utility.** Electronic mail offers new functions. For example, a new distribution list can be appended to a message you have just received, and secondary distribution accomplished in a matter of minutes. Detailed audit trails can be developed to show when each recipient received the message and when he *acted on* the message.

Electronic Filing

Once a document exists in digital form, it can be stored and retrieved electronically. Because the document can be retrieved over electronic communications links, the storage location does not limit access. Rather than keeping copies of the documents in a local computer system, an individual needs to keep only an electronic list of the names of the documents. Additionally, electronic files can have electronic indices that provide access to the documents by author, title, keyword, etc. Electronic filing has many attractions.

- **Convenience.** Electronic files will be easier to use than most paper files.
- **Completeness.** Individuals can access much larger information collections.
- **Utility.** Additional functions are available. For example, a document referenced in a memo can be accessed almost instantaneously.

PICTURES

Document transmission and storage is not limited to text. Facsimile equipment (FAX) that exists today digitizes pictures for transmission to remote locations over communications equipment. The efficiency of FAX transmission has consistently improved, as more complex digital processing is used to reduce the number of bits needed to transmit a picture. For example, a typical illustration is predominately white space. If the white space can be encoded effectively for transmission, the total volume of data can be greatly reduced. Over time, more and more complex electronic circuits are used, and greater coding efficiency is achieved.

Computer terminals in the future will include graphical display capabilities. In the past, drawing pictures on a terminal required special expensive circuitry. In the future, graphics will be drawn by inexpensive microprocessors and integrated circuit memory.

- The terminal will include a large amount of IC memory that will be used like a piece of graph-paper for drawing pictures – one bit of memory for each dot on the picture. A grid of 400-by-400 dots requires 160,000 bits of memory and provides picture resolution comparable to a television set. More bits provide greater resolution.

- The microprocessor will use this memory to create pictures, by setting and clearing bits representing light and dark in the picture.

Computer printers will typically be able to reproduce pictures as well as text. Offices will probably include a copier-like paper-to-data conversion device that will perform these functions:

- **Picture digitizing.** It will convert a drawing to a standard digital format. This is the function of a facsimile transmitter today.

- **Text identification.** If the "drawing" is text (for example, a typewritten sheet), the picture data will be compressed to the more highly encoded text form; the letter "a" will be converted to the code for "a," which uses far fewer bits than a digitized picture of the letter "a." This is the function performed by an OCR reader today.

- **Printer.** The device will also perform the inverse process, turning digital-form data back into pictures and text.

TOTAL INFORMATION SYSTEMS

The net effect is that the computer system disappears as special, visible technology. Before the computer, data was created, transmitted, and stored in a standard form, namely paper. The computers have introduced an abundance of special information processing devices. But eventually, improvements in technology will lead to computer equipment capable of storing, transmitting, and displaying most of the data that has traditionally been paper-based.

The resulting information systems offer greatly extended functional capabilities compared to the manual, paper-based predecessors. In the future we may expect more homogeneous systems capable of performing storage, retrieval, processing, and transmission tasks.

SUMMARY

A speculative glimpse of the future has been presented. The existing level of information automation is primitive. Continuing improvments in technology and full utilization of existing technology will lead to many future developments.

Chapter 15
DISTRIBUTED SYSTEM APPLICATIONS

So far the discussion has been about distributed information systems at a very general level. As a practical summary, this final chapter describes some diverse, practical applications of distributed computer systems. Appendix A includes case studies based on actual computer systems.

LABORATORY SYSTEMS

Minicomputers are an invaluable part of medical, scientific, and industrial laboratories. A standard for instrument/computer interconnection has been developed (IEEE-488) and is incorporated in many instruments. Use of IEEE-488 standard equipment permits instruments from many manufacturers to be connected to a single computer system to provide program-controlled instrument set-up and data acquisition. Setting up a complex experiment may be as simple as getting the needed instruments from a stockroom and cabling them to the lab computer.

Although the cost of a powerful minicomputer processor and central memory system is quite low, many useful peripheral devices are still relatively expensive. Resource-sharing distributed systems provide small laboratory computer systems with the functional capabilities of expensive peripheral devices, such as large disk drives, printers, and plotters, at a fraction of the cost of the device.

In a resource-sharing distributed system, the satellite computers used for experiment control and data acquisition are simple; they consist of a processor, central memory, and a network communication link. A larger minicomputer system is included in the network, which includes the expensive peripherals. Program development and data reduction are primarily performed on the large minicomputer. The satellite computers use the central system for data and program storage over the data network, and for special services such as data plotting.

Distributed system technology can also be used to interconnect remote laboratory facilities. The remote laboratory can be a satellite lab, or it can even be unattended instruments.

Drug companies use networks of computers to monitor and record the progress of laboratory animals that are used to test the safety and efficacy of newly developed drugs. Extensive data is gathered, often from more than one testing site, and analyzed in order to generate reports to the regulatory agencies.

Airports that are near residential communities have developed instrument networks for monitoring the noise of airplane traffic using the airport. The network consists of a set of remote sound-measuring devices that are connected to a central minicomputer. Similar systems have been used for the real-time measurement of air pollution in different areas of a city.

MEDICAL INFORMATION SYSTEMS

Health care in hospitals depends on information processing. Diagnostic and therapeutic data storage and retrieval are essential parts of health care. A large amount of information processing is needed to schedule hospital facilities (e.g., beds, X-Ray equipment), to bill for the services rendered, and to deal with third-party payers (e.g., insurance companies, health-care plans).

A continuing problem in the design of hospital information systems is the diversity of processing styles and requirements involved. It has proven impossible to develop a single computer system that meets all the different needs. One approach is to use departmental minicomputers, each programmed to meet local needs, interconnected into a distributed system.

The following examples give some idea of the breadth of requirements.

- **Admitting and Administration.** These conventional data processing applications adapt nicely to interactive, forms-based, transaction processing on a minicomputer.

- **Accounting.** Medium-to-large hospitals generate large volumes of accounting data. A batch processing accounting system with some interactive query capability serves nicely here.

- **Pathology.** The clinical laboratory is a standard, small laboratory application.

- **Surgery.** A surgery department has some scheduling problems that might be computerized, but effective word processing systems for preparing surgical reports is a major need.

PLANT MANAGEMENT SYSTEMS

Machine and process control were early applications of minicomputers. For example, a numerically controlled milling machine is capable of precisely machining a complex part with a minimum of manual intervention. In process control, early minicomputers served as process monitors by repeatedly measuring many process variables, comparing the measured values to preestablished limits, and sounding an alarm if out-of-limit conditions were found.

Machine and process control have become more sophisticated over time. Systems exist that machine parts from engineering drawings which are also prepared on a computer. Process control systems now automatically control the process as well as perform monitoring functions.

Computers perform valuable management and cost accounting functions within a factory. Time reporting and attendance data is gathered by an interactive computer system with sturdy terminals located throughout the plant. These terminals serve a time-card function by recording the hours worked for each employee in a machine-readable form that can be directly input to other data processing systems, such as payroll and project management.

The same terminals are also used to follow particular jobs through the manufacturing process and gather detailed cost accounting data. The same data can be used to study machine utilization and reliability and can serve as input to scheduling programs.

These various computer systems can be joined into a distributed, plant management system by using network technology. Interconnecting the systems has these benefits:

- *Data is available more quickly.* Problems can be seen and solved in timely fashion.

- *More data is available.* Larger processes can be optimized; for example, work flow through the plant can be considered as easily as work flow at a specific machine.

The resulting plant management system can be tied into a corporate distributed system that includes systems in other plants, warehouses, and sales offices.

PUBLICATION

The preparation of this handbook is an example of the use of distributed systems. The text was composed and edited on a small word processing system (a DIGITAL WT/78). Most drafts were printed on an attached LA180 DECprinter I dot-matrix printer. Major drafts were printed on a "daisy-wheel" printer attached to a larger WPS-102 word processing system. Text was moved between the two systems by interchanging floppy disks, but telephone communications could have been used, since software exists to interchange documents.

The finished text was input automatically into a DIGITAL DECset-8000 typesetting system. A translator program read the WPS floppy disks and translated them into a composition format. The DECset-8000 system includes many typesetting features that are not part of a word processing system, such as selection of type fonts and sizes.

The same typesetting system receives input from other word processing systems, and from on-line composition terminals. These terminals are specially engineered for copy preparation and typesetting. Each terminal has a small computer processor internally, so the typesetting system is a distributed system by itself.

Computer programs perform the traditional printing task:

1. Characters are added to a line until there is no room for another complete word.

2. The width of the resulting line is measured.

3. If the width is less than a prespecified limit, a hyphenation program is used. The hyphenation program uses grammatical rules and tables of special cases to discover where the next word can be hyphenated. The largest possible piece is added to the line to fill the blank space left.

4. The resulting line is justified. Small spaces are added until the line is flush with the desired right margin.

5. The text and spacing data is used to drive an on-line typesetting machine that exposes film with precisely positioned and sized letter images.

The generated film is developed into galley proofs that are cut-and-pasted into the printing masters for the book.

Many larger newspapers use similar systems for printing the paper. A newspaper system typically uses a larger computer for final composition and typesetting because of the complexity and time constraints of producing a daily paper. The same computer can also do general business processing for the newspaper, as well as specialized processing such as classified advertising management.

FINANCIAL WIRE SERVICE COMPUTERS

Banks, and other financial institutions, use data processing and communications extensively. In addition to whatever systems a bank may develop for its own use, which are often distributed systems, there are common "wire" services that transmit messages between banks. The wire services are used to transfer funds between banks, as well as to perform other communications.

Historically the messages have been sent and received manually by teleprinter operators. Recently, banks have installed minicomputer systems that attach directly to the wire services, and automate the message functions. These minicomputers are connected in turn to internal computers and terminal networks.

These wire service computers speed up message processing, and permit more sophisticated use of interbank transfers. For example, funds can be moved around to take advantage of maximum short-term interest rates.

The value of adding a computer to perform message switching is evident from the magnitude of funds involved.

- A large bank will process in excess of $10 billion ($10^{10}$) in transactions daily.

- At an annual interest rate of only 10 percent, the value of one day's interest on $10 billion is roughly $3 million!

RETAIL MERCHANDISING

Most large retailers are installing computerized point-of-sale (POS) systems. A POS system begins with a device that combines the functions of an intelligent computer terminal and a cash register. The terminal often is able to read merchandise labels automatically. The POS terminal performs the traditional cash register functions of accumulating the total sale value, computing sales tax, and producing an itemized receipt and register journal.

The POS terminals typically connect to a minicomputer located on the store premises that has on-line data storage. The store computer can perform credit authorization for store accounts (or connect to a remote credit authorization system for major credit cards). The store computer keeps a record of all sales in the store.

The store computer often connects to warehouse computers, and corporate accounting systems. By use of the distributed system, the store management is able to track sales trends closely and thus to react to merchandising problems and opportunities quickly. Similar use of the data can be made at the corporate level.

MANAGEMENT SCIENCES APPLICATIONS

Larger companies often use computer programs for economic modeling and other planning and analysis activities. The programs are often designed to be used interactively, so that various different hypotheses or strategies can be conveniently tested and compared. Typically the programs have been developed using an external, timesharing service.

With the availability of powerful and easy-to-use minicomputers, many of these planning applications are being moved back in-house. Minicomputers are powerful enough to run the same programs, still with all the benefits of interaction, but at a substantially lower cost.

Such a planning minicomputer can be tied into various company information systems so that recent operational data is available online. An additional benefit of the in-house minicomputer approach is security: sensitive data doesn't have to be stored at a computer service that other businesses use, including competitors and potential competitors.

INSURANCE CLAIM PROCESSING

Claim processing in an insurance company home-office is an ideal minicomputer transaction processing application. A dedicated system can be implemented for each major department. The system is used for

- entering new policy data
- entering claim data

- producing policy and claim documentation

- on-line queries regarding policy or claim status

Using departmental computers instead of one large system facilitates adapting each system to the particular style of work. Workman's compensation processing doesn't have to fit the same pattern as life insurance processing.

The departmental system can be connected to other corporate data processing systems.

SUMMARY

The final chapter has presented a diverse set of distributed system applications in order to make the potential of distributed systems more obvious. Additional examples that have been abstracted from existing systems are given in Appendix A.

Appendix A
CASE STUDIES

A TAXONOMIC MODEL

The benefits of distributed data processing and the criteria for choosing it can be highlighted by analyzing several recent installations. Although each case has unique parameters (and therefore represents a singular response to a particular set of needs), the common theme running through most such implementations is the flexibility afforded in the distributed processing environment – flexibility to optimize the utilization of hardware, software, and communications resources for individual tasks and sets of applications within the scope of a product.

Distributed data processing is a broad concept encompassing a number of different hardware, software, and communication disciplines. Implementations are not configuration-bound; the same equipment can support different computational and data structures simultaneously or can serve as the foundation of a dynamically evolving operation. One classification of possible structures for distributed applications is a taxonomy as illustrated by Figure A-1.

Type A – Stand-Alone Systems

The earliest electronic data processing (EDP) facilities were stand-alone machines, each characterized by a single processor accessed from one batch or interactive terminal. Stand-alone systems are manifestly undistributed. This classical approach to data processing is still viable for certain classes of applications – modern examples being the packaged WS 78 Word Stations and the Datasystem 310 small business computers by Digital Equipment Corporation.

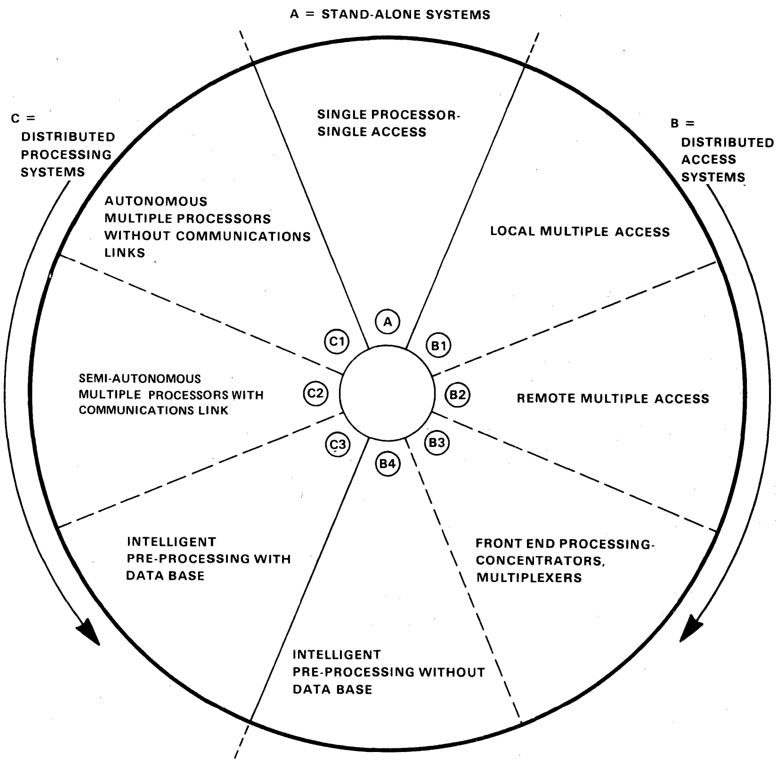


Figure A-1. A taxonomic model of distributed applications.

Type B – Distributed Access Systems

A single processor or computer can be accessed simultaneously from multiple terminals in batch, interactive, or mixed operating modes. The objective is to exploit the high speed and storage capacity of state-of-the-art machines, making resources available to users at individual work locations while maintaining central processing capabilities and a single integrated data base.

Type B-1 *local multiple access* systems have all terminals in close proximity to (and wired directly to) the central processor. The resources are distributed through some form of time-sharing executive in the operating system, which handles tasks such as assigning terminals to jobs and resolving contentions for memory. No separate equipment or procedures are needed to manage the communications.

Type B-2 *remote multiple access* systems have at least some terminals geographically distant from the central location. Various forms of communication lines and disciplines are needed to establish connections between the remote terminals and the processor, and to perform error checking and security functions.

Remote Job Entry (RJE) batch stations, interactive time-sharing utilities, and shared transaction processing are examples of Type B-1 or B-2 systems, depending on the siting of the terminals.

Type B-3 systems use various forms of *concentrators, multiplexers,* or *front-end processing,* with remote multiple access terminals, to reduce the capital and operating costs associated with communications. Some hardware and software logic is used in the field, but the tasks performed involve communication only and are invisible to the data.

Type B-4 sysems have capabilities for *preprocessing without remote data base.* The purpose is to provide limited on-site checking of transactions (for instance, for format or quantitative limits) to minimize use of communication lines for erroneous data.

Type C – Distributed Processing Systems

Applications can be segmented for execution on multiple processors, some or all of which may have individual data bases. The objective is to place actual computing power and control of data bases near appropriate work locations and to permit some or all tasks to be performed locally, to avoid unnecessary communications overhead.

Type C-1 *autonomous multiprocessor* systems are suitable for applications that can be divided among independent processors and data bases, with no interconnecting communication links. An example would be a series of minicomputers dedicated to controlling and monitoring separate units in a chemical plant.

Type C-2 *semi-autonomous multiprocessor* systems have the computing load and data base for a unified application shared by several processors. Communication links are maintained between processors and used for coordination. Transactions may be handled at a local processor or sent to one or more remote machines for disposition. Network topology may be influenced by any number of factors – hierarchical structure of an organization or optimum loading of equipment being two common illustrations.

Type C-3 *intelligent preprocessors* with data base configurations involve substantial on-site handling of transactions, including direct access to local files. Most transactions are processed on-site, but some may be sent elsewhere in the network, as appropriate.

I. GOVERNMENT AGENCY SUPPORT

A federal agency was formed about ten years ago to gather and analyze data as a means of recommending national standards, enforcing existing regulations, monitoring certain local industrial and municipal activities, and performing research and development functions. The agency currently has over 11,000 employees. Approximately 3,500 are at the Washington headquarters; the remainder are assigned to ten regional offices, several research centers, and a number of laboratories throughout the United States.

EDP Applications

The agency has substantial EDP requirements. These include performing administrative tasks and executing scientific functions as well as building, maintaining, and providing access to large national data bases. Representative applications at the national level include programs for contract, library, and resource management;

maintenance and manipulation of data bases and models; information collection and distribution; and supervisory control. There are also administrative and scientific applications unique to individual sites. Due to the geographical dispersal of its activities and the need to offer computational services to state and local governments, the agency supports resources for nationwide access to its programs and data. At present, 2,500 individuals use the facilities - 333 of whom can be on-line simultaneously.

Original Approach

In the original system evolved by the agency, about 70 percent of the total EDP load was implemented on a time-sharing utility under contract to an outside vendor. Of this usage, roughly 70 percent was for the national application programs and 30 percent supported regional and laboratory tasks. The installation was an IBM 370/168, accessed by a vendor-maintained telecommunication network of remote job entry (RJE) and interactive terminals. Twelve agency sites with heavy concentrations of users had dedicated multiplexers; other locations accessed the computer by WATS lines, the Federal Telephone Service (FTS), or the public switched network. This configuration would be termed a remote multiple access system (Type B-3) with multiplexers to reduce communication costs; it represents a *distributed access approach* to EDP.

An in-house Univac 1110 computer carried 20 percent of the total EDP load. This installation also supported RJE and interactive terminals accessed through FTS, WATS, and the switched telephone network. This is a remote multiple access system (Type B-2).

The remaining 10 percent of the applications were run on resources belonging to other government agencies or to outside contractors. Communications were implemented in various ways.

Deficiencies in the Original Approach

Most of the national applications were written as batch programs, with data keypunched and verified before being input to RJE terminals. This created substantial delays in the processing and distribution of information. In order to minimize delays, the agency encouraged conversion to interactive operation.

The conversion from RJE to interactive operation substantially raised the load on the two major computer resources causing responsiveness to diminish noticeably. For example, during a survey period in FY76, only 72 percent of the top priority jobs began execution within the requested time. Compounding the response problem, internal EDP requirements were growing by nearly 30 percent per year, further degrading responsiveness.

Budgets allowed only 2.5 percent increase in expenditures for data processing operations (in spite of the 30 percent projected usage growth). With the original system configuration, service simply could not be expanded to meet the expected demand under this monetary constraint.

Another problem was that each RJE terminal could be connected to only one host - the 370 or the 1110. For dual access, it was necessary to disconnect a unit from the network manually, to dial the other processor, and to stay connected until a task was completed. As a result, many sites needed two such terminals to run the job load.

Requirements for a New System

The agency decided to investigate means of meeting future EDP requirements. Four major criteria were established for evaluating proposed approaches.

- Cost was of special concern because of the disparity between projected EDP growth and budget increases. The incremental cost for EDP resources would have to be reduced substantially, or it would be necessary to limit the processing functions at headquarters and at field locations.

- Increased responsiveness was a major criterion. Due to the rapid growth of EDP and to the transition from batch to interactive processing, turnaround time was becoming unacceptable. The problem was especially acute in the case of rapidly changing data.

- Maintenance of management control from headquarters was considered critical in order to maintain uniform policies and procedures throughout the country.

- Minimal disruption (when converting from one system to another) was also essential because many ongoing programs could not tolerate outages. Further, it was recognized that any disruption of the central EDP function would have adverse impact throughout the organization.

Alternatives

The first approach considered was the expansion of existing centralized facilities. This approach would involve doubling the number of ports on both systems over a period of two years and installing additional RJE and interactive terminals at field locations.

- An expanded centralized multiple access system would be minimally disruptive because it would entail little or no change in the operation of existing national programs.

- Cost was unattractive. The agency estimated it would incur an annual expenditure increase of nearly 30 percent to match the rise in usage.

- From the standpoint of management control, the agency determined that the centralized facility would be advantageous because management of the EDP resources would be concentrated at the Washington headquarters.

- Finally, the responsiveness of the system, already poor, would grow even worse with an expansion of this magnitude. While additional central processors could be obtained, this would further escalate costs.

The second approach considered was an autonomous multiprocessor configuration with independent minicomputers placed at a number of major regional locations. The minicomputers would not be connected to the central resources, but would offload the areas of largest growth (regional and R&D processing) from the main systems.

- The major benefit would be improved responsiveness, but migration of applications from the central to the regional systems would require time, and there would undoubtedly be further degradation of performance during the conversion period.

- Disruption could be minimized if existing terminal equipment were kept on-line for a reasonable period after the minicomputers were installed.

- The management control criterion presented some serious problems. Enforcing central management standards would be difficult because the autonomous processors would likely evolve differently under independent regional supervision.

- The cost of the autonomous multiprocessor approach was deemed high. Existing terminals and new minicomputer equipment would be maintained concurrently during the transition period. In addition, the delay between the installation of the minicomputer equipment and the time when regional processing could be substantially offloaded from the central computing resources would increase utilization expenses for a long interval before any savings could be realized. Moreover, to support the national application programs, some terminal equipment might have to be maintained indefinitely.

The third alternative involved employing minicomputers as semiautonomous multiprocessors with communication links to the central resources. The concept included using emulation programs at the regional locations to permit the dispersed minicomputers to smoothly assume the functions previously performed by the local RJE and interactive terminals. In this manner, a transition period could be established during which existing applications would be run on the minicomputers with no programming changes. The conversion of regional programs from the central computer to local operation could proceed in parallel with this transition.

- The disruption caused by this approach was projected as substantially less than would be incurred by the use of autonomous minicomputers. A pilot site would be used for debugging the emulator software.

- The agency predicted that the degradation of central system responsiveness would be halted with this approach.

- Agency personnel also thought that management control of the regional operations could be achieved using the communication links to establish and enforce national standards on local operations.

- Finally, the cost benefits of the semi-autonomous minicomputer configuration appeared significant. Preliminary analysis indicated that having the minicomputers take over the terminal applications as well as local processing would substantially lower system costs and provide the needed incremental growth capabilities.

The Approach Taken

The agency selected a semi-autonomous minicomputer-based system. The overall configuration is shown in Figure A-2. This concept appeared to offer the best combination of responsiveness, minimum disruption, management control, and cost reduction. In addition, it could provide each region with the ability to set its own priorities and maintain better control over the timeliness of its data.

To encourage standardization at remote facilities, the agency specified a minimum configuration for each location. This was built around the Digital Equipment Corporation PDP-11/70 processor (Figure A-3). Each local node has three ports, as shown, comprising leased lines to the central 370 and 1110 plus a dial-out interface to other processors and for backup use.

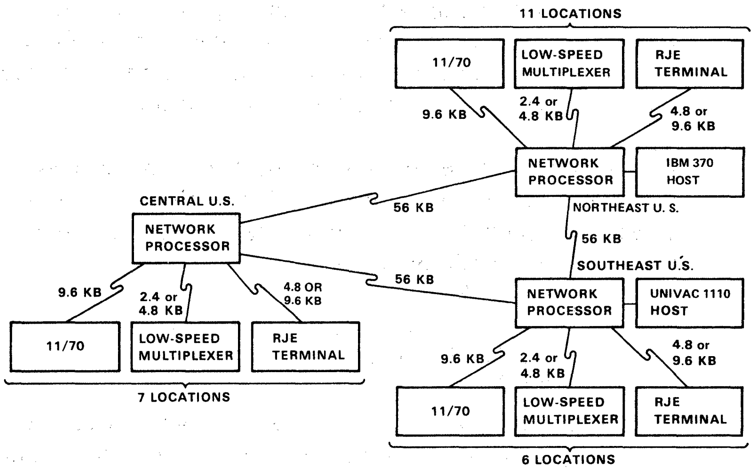


Figure A-2. Simplified communications network.

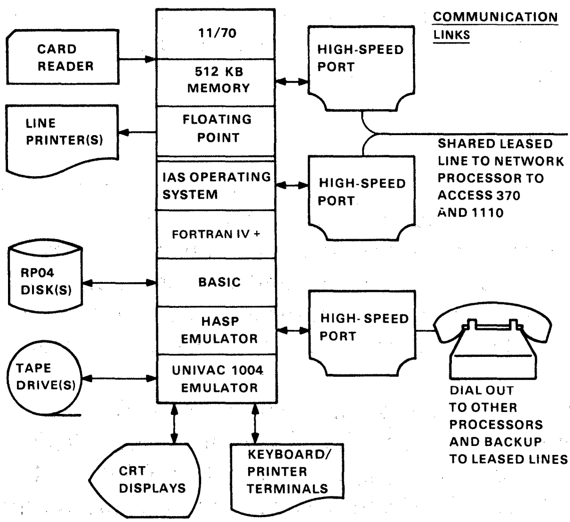


Figure A-3. A typical RJE/interactive mode.

After establishing this uniform hardware and software configuration and developing a standard contractual arrangement with DEC, the agency invited each region to do a cost-effectiveness study for the installation of such a system in its application. The result would permit each location to decide whether to replace existing RJE and interactive terminal equipment with PDP-11/70 equipment.

Three regional offices, two research centers, and one laboratory have already completed feasibility studies. In each instance, they found the cost of the distributed multiprocessor installation to be justifiable. The studies also showed that minicomputers would allow the regions to offer better service in the future. In most cases, the capital investment in the systems over their useful lives would be significantly less than rental of the existing terminal equipment.

To ensure minimum disruption during changeover, agency data processing specialists developed a uniform staged installation procedure to be followed in each region. During the first phase, all current applications would be transferred unchanged from the RJE and interactive terminal equipment to the minicomputer, exploiting the emulation of the original programs. Cost savings would be realized immediately by eliminating the rental of the existing terminal equipment and by reducing connect time due to more intelligent local processing of transactions.

During the second phase of implementation, a number of local applications would be transferred from the central computer – to be run on the minicomputer alone.

During the third phase, the regional offices would cooperate to offload some national application programs from the central computers to the regional systems. Many applications lend themselves to a mixed mode of processing in which the bulk of the work can be done locally, with the central computer used only for integrated data base management and interregional coordination.

During a possible fourth phase, a fully distributed approach could be adopted. All applications would then be transferred from timesharing vendors to a combination of the in-house Univac 1110 and the remote PDP 11/70s. Because this would require headquarters initiative and direction, this was to be deferred until conversion was well along at a number of sites.

Benefits

At present, five locations have converted from RJE and interactive terminal equipment to the PDP-11/70 configurations. As additional field organizations complete cost/benefit analyses, they are expected to adopt the new systems. The pilot PDP-11/70 was operating successfully as an emulator within two months of installation. Subsequent installations were accomplished even more rapidly.

All installed systems are now being used with RJE emulation. Cost recovery has started with the removal of the old leased terminals and the transfer of some processing from the central computer facilities to the PDP-11/70s. Increasing economies are expected to be realized as the conversion process proceeds through secondary and tertiary phases. Improved responsiveness at the central computer site is evident already.

Because the local laboratories and research centers are no longer limited by the central system, they are reporting improved operations with priorities under control. This enables the regions to better serve their own needs, especially in disseminating information to local and state governments.

The PDP-11/70 configurations enhance the conversion of former RJE applications to interactive usage. As applications are switched from remote to local operation, there is faster turnaround of information and more timely reporting for operational and R&D requirements.

Overall experience of the agency with distributed processing has been positive to date. Careful planning and recognizing the importance of managerial controls from the beginning have permitted the agency to achieve the conversion from centralized processing with maximum benefits.

II. AUTOMOTIVE PARTS AND VEHICLE DISTRIBUTION

A large automobile manufacturer found that the distribution of parts and matching of vehicles with orders had become a significant electronic data processing (EDP) application. The firm stocks 60,000 different parts distributed from seven depots in North America. The company also maintains records of all automobiles in inventory dealership, so that customer orders can be filled from existing inventory rather than assembling. Because the number of parts and volume of transactions is great, management considers EDP essential to optimizing stock and transfer policies.

Original Approach

The firm had evolved a data processing system with several disparate elements. Each of these components was developed somewhat independently to meet particular sets of needs.

Perpetual inventory at each parts depot was maintained on Cardex files. This would be a Type C-1 stand-alone distributed configuration under the taxonomic model, though it is not strictly an EDP system. Manually generated summary reports were sent to distribution headquarters although no attempt was made there to perform centralized inventory control. At each location billing functions were provided by small business computers operated in a batch mode using punched cards. Because no communication was maintained between billing systems at individual depots, this part of the total EDP function can be classified at Type C-1.

Automotive searching (determination of car inventories) requirements were met by a central IBM 370 computer at distribution headquarters. This machine used a number of local interactive terminals for changes and interrogations (Type B-1 configuration). Dealers accessed the data base by telephoning their zone offices which in turn relayed the requests by Telex to system operators at headquarters. The reverse procedure was used for relaying responses. Typically, the zone offices are located at the parts depots.

Deficiencies in the Original Approach

The major shortcoming of the original approach to parts distribution data processing was the lack of overall control, with consequent expense of overstock and, conversely, delays due to unavailable items. The Cardex system was adequate for incrementing or decrementing inventory on hand, but it did not include provision for locating parts at other depots or means for discouraging one depot from hoarding items needed by others.

The small business computer likewise satisfied local billing needs, but there was an obvious duplication of effort in entering essentially the same data into both the stocking and the invoicing systems. Moreover, the computer produced little routine management information, and reports were printed documents rather than files that could be manipulated and interpreted automatically by a centralized machine.

The automotive searching system was also limited in effectiveness. Telephone and Telex costs were high, and maintaining a team of operators to handle requests was expensive. The delay in obtaining the information and the opportunity for error due to the number of steps involved were additional difficulties.

Requirements for a New System

The automobile manufacturer set a high priority on unifying EDP applications. The broad objective was to provide a management information system that would give regional depots better control of local operations while permitting headquarters to supervise the overall organization more effectively. One specific goal was to lower distribution costs by using a centralized inventory control and to reduce labor-intensity by using a source data capture procedure. Another goal was to cut the costs and delays associated with vehicle searching. A third need was to facilitate communication throughout the organization.

The company studied numerous options, and seriously considered two approaches. Because of the inherently dispersed nature of the organization, both approaches involved forms of distributed networks. Management then examined the following criteria:

- Capital costs, including the time and effort level required for programming, and the risk factor associated with development

- Operating costs, in particular the manpower and communications expenses

- Overall system availability and the effects of equipment failure at single locations

- The integrity of data carried from point to point

- The provision for regional control over operations – subject to enforcement of EDP standards and need for uniform central supervision.

Alternatives

One proposal was to expand the capabilities of the existing IBM 370 and make the resources available through terminals at the depots. This would be a *distributed access* system classified as Type B-2, B-3, or B-4, depending on the extent and functionality of preprocessing available at the depot locations.

- This approach offered advantages in terms of capital costs. Although the central processor would have to be expanded, the investment in the terminals for the remote sites was relatively modest. Staff members were familiar with the software requirements on the machine, so programming was not expected to pose particular problems, and development was not considered to be too risky.

- Operating costs were expected to be moderately high. Personnel requirements for operating the system through the terminals were reasonable, but the communication network necessary to provide real-time access and high data integrity would be expensive to maintain.

- Availability was considered a serious deficiency of this approach. Although the processor was reliable, an outage would bring down the entire system.

- Data integrity, as suggested, could be made high, but at the price of communication network complexity and cost.

- Regional control was also problematic with the distributed access approach. Managers at each depot would be responsible for local operations but in this system would lose control over their data and over the operation of the EDP function. Uniformity, however, would be virtually assured.

The other alternative considered was a *distributed processing* system. Semi-autonomous multiprocessors with data base support would be located at the depots, and a supervisory machine (also having a data base) would be used at the headquarters location (a Type C-2 system). Within the node at each depot, multiple access would be provided for real-time data entry and retrieval (a local Type B-1 configuration). The data base for car searching would be resident at the central location but could be accessed from the terminals at the depots, with the local processors used for transaction checking (a Type B-4 mode). The entire network would be exploited for message transmission (for this function, a Type B-3 system).

- The autonomous multiprocessor approach meant a larger capital investment for the automobile manufacturer because moderate-capacity computer systems with mid-range mass storage systems would be necessary at each depot and at headquarters. The PDP-11/70 computers being considered, however, could be specified with business-oriented software that permitted programming of all functions – including communications interfaces – in the DIBOL language. This software was considered relatively simple and eliminated the need for producing system-level code, so programming effort would be modest and development risk low. The system would use DECnet/E networking protocols between the PDP-11/70s. A proven 3271 emulator package for the headquarters PDP-11/70 would allow direct interaction between this machine and the 370.

- Operating costs were estimated to be quite reasonable with this approach. The data checking protocols in DECnet are implemented in the processors, reducing the need for sophisticated communication lines. Maintenance of local data bases would permit most of the real-time functions to be performed on-site; it would also allow much of the high-volume message traffic to be carried at night in a batch mode making line usage relatively low. The manpower requirements were also reasonable. Placement of processors and mass storage at the depots would not require hiring of local data processing specialists, because practically all systems work could be performed at headquarters and downloaded through the network. Moreover, operation would be simple enough for regular clerical personnel to use the terminals directly.

- System availability was another strong point of this approach. Failure of the processor or problems with the data base at any depot would not affect operations elsewhere. A failure at headquarters would allow inventory management, the most time-critical work, to proceed at remote sites. Long-term failure at headquarters was unlikely because the proposed system would include a pair of PDP-11/70s to separate development from production work and at the same time would offer redundancy for backup.

- Data integrity was also outstanding because of the checking functions inherent in the network architecture.

- A high degree of regional control was inherent in the distributed multiprocessor approach. Questions about enforcing standards were raised, but because each depot depended on systemwide information for optimum operation, incentives would be strong to adhere to established procedures, even for special applications. Further, the availability of competent programming resources at the headquarters would minimize the desire of depot managers to perform system work outside the established set of constraints.

The Approach Taken

After evaluating the criteria, the automobile manufacturer elected to proceed with a distributed processing system. The firm believed that this would yield not only the most cost-effective implementation of current tasks, but the greatest flexibility and the clearest paths for projected growth.

User personnel, working with representatives of the computer supplier, designed a configuration in which each of the seven regional depots has a DEC PDP-11/70 processor with three disk drives and six cathode ray tube (CRT) terminals. Three of the CRTs are projected to be needed for on-line inquiries; the others are expected to be used for entry of receipt and order information. Collectively, they should handle about 2,000 order-line transactions daily. The hardware and software installation for a representative depot is shown in Figure A-4.

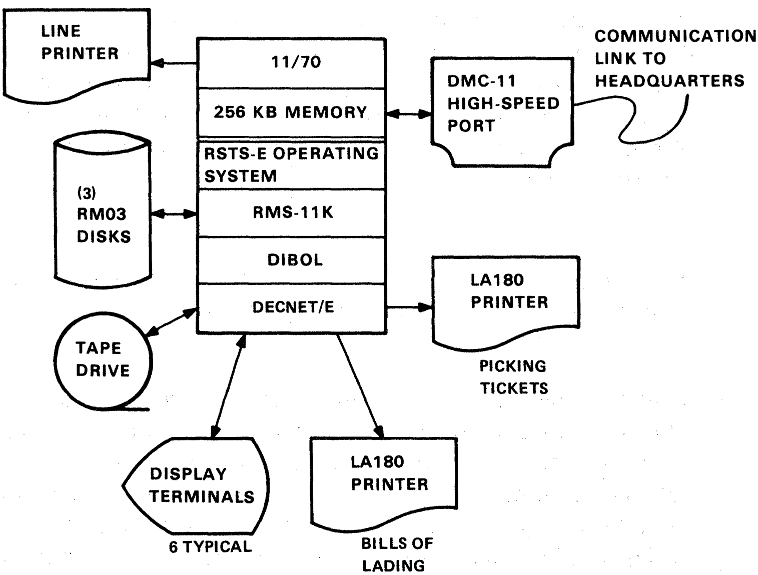


Figure A-4. A typical depot configuration.

The headquarters location includes two identical PDP-11/70 processors each having a dedicated disk drive with access to a second disk through a switch, shown in Figure A-5. The PDP-11/70s include 3271 emulator packages which permit bidirectional communication with the 370 using IBM CICS transaction communication procedures in a manner transparent to applications on both machines.

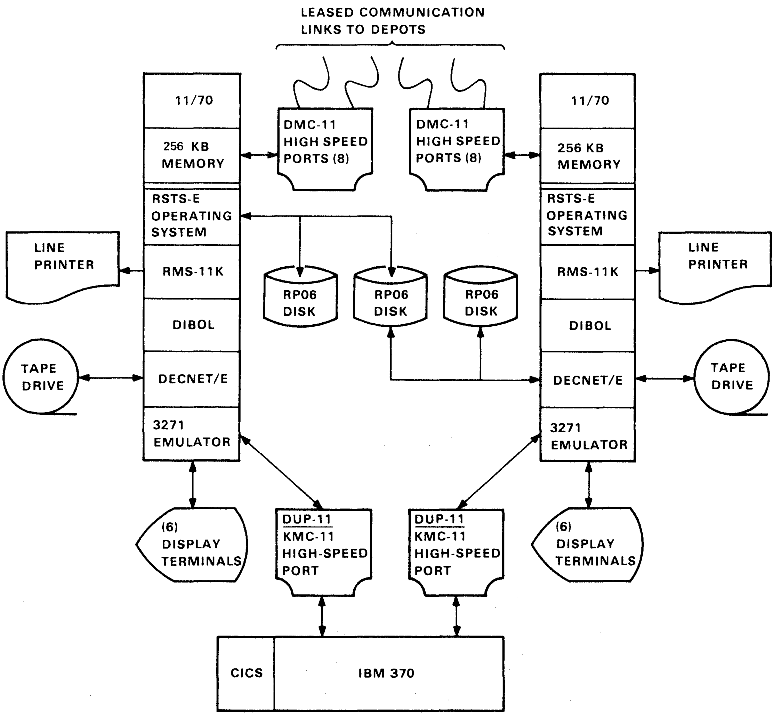


Figure A-5. Headquarters configuration.

DIBOL commercial programming language is employed for all tasks on the PDP-11/70s, including intermachine communication. The programs were all written by the user; tasks are run under the

RSTS/E resource-sharing time-sharing executive; DECnet/E manages all communications in the PDP-11/70 network; and the 3271 emulator interfaces the headquarters PDP-11/70s to the 370.

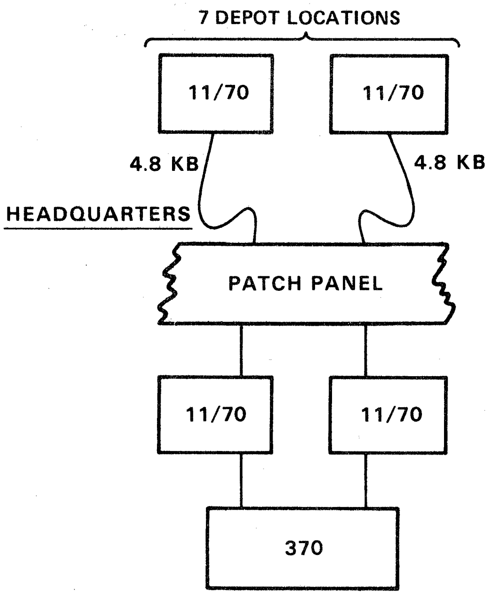


Figure A-6. Communication network.

Communication is implemented in a star network, with the headquarters location at the hub and the depots at the periphery (Figure A-6). The dedicated 4800-baud communication lines converge in a patch panel at the headquarters to permit switching between the redundant processors.

Parts inventory management was the first class of applications implemented. A complete inventory for each depot is maintained locally and updated in real-time through the multiple access terminal subsystem at this depot. An inventory file for each of the other depots is updated every night by a transaction list sent from the headquarters system.

Parts are logged through the terminal upon receipt at the depot. The system increments lists, assigns warehouse locations, and generates confirmation and other documents. Likewise, when parts are requested, the order is entered at a terminal. If the item is in stock, the system decrements the inventory list, produces a picking ticket, and generates all necessary billing and ancillary papers. When a part is unavailable, the system creates a backorder transaction, searches the local copies of other depot inventories, and returns an appropriate message to the operator. Assuming that the system locates the part at another depot, the operator can enter a transfer request. This message is sent to the headquarters system in real-time and relayed immediately to the selected depot. If the second depot agrees to fill the order, it initiates a response confirming the transaction and cancelling the backorder. If the interrogated depot is not in a position to supply the part, it sends a refusal message. The originating station can then try another depot or let the backorder be processed through the system.

To minimize communication burdens, each depot transmits its list of demand and sales transactions in batch mode at night. The central PDP-11/70 then updates the master inventory. Headquarters, in turn, sends a complete list of inventory changes to each depot, allowing the local processors to update their files for all the other locations.

The automobile searching task is the second application to be implemented. The distributed system provides direct access to the central data base from a terminal at a regional sales office. The depot PDP-11/70 simply acts as a message preprocessor; the central PDP-11/70 channels the communication directly to the 370, all in real time. The result is that the dealer can get an answer while talking on the phone to the regional sales office.

Benefits

Implementation of the inventory and car searching applications on the distributed system is expected to free an average of 300 man-hours weekly at each depot for other tasks and to significantly reduce the telephone, Telex, and mail costs. These efficiencies, coupled with the reduction in stock and delays, more than justify the capital and operating costs.

System availability has proved to be outstanding, with only minor delays experienced in accessing files for updating or order processing at individual depots. No problems affect more than a single location. Integrity of data has also been outstanding, with the error-checking features of DECnet preventing the propagation of false signals through the network.

Regional managers have expressed satisfaction with the degree of control they believe they exercise over their data bases and with the consequent ability to better manage the inventories at their depots. Customer delays due to backorders have been reduced without any complaints about usurpation of authority to decide whether parts should be shipped to other depots for ultimate disposition. At the same time, the data processing staff at headquarters believes that goals for enforcement of EDP standards and uniform central supervision are being adequately met.

At the current level of activity, the originally contemplated applications will use only 20 percent of the installed system capacity. As a result, additional tasks will be added. Plans are for horizontal as well as vertical expansion, with new applications being devised while the functionality of present tasks is increased.

The warranty system, a data base which indicates vehicle ownership, is an example. By unifying this information, the company expects to simplify the huge job of updating records when cars change hands, to provide quick access when questions arise, and to lower the costs associated with contacting owners for reminders about service needs or recalls.

The distributed processing approach selected also gives the company the advantage of a low-cost growth path. Even with expansion of existing functions and addition of new tasks, the established base of hardware and communications is expected to meet demands for the next five to ten years without major enhancement.

III. RAILROAD YARD MANAGEMENT

A large midwestern railroad company operates more than 20 major and 130 secondary freight yards. Trains arriving at these terminals are disassembled for car reassignment. Some cars stay at the yard for unloading and reuse or repair. Others are reassembled, along with those originating at the yard, into new trains.

The company has strong motivation for handling cars quickly and efficiently in the yard. Throughput is important because goods may be perishable or have time value; the company pays a per diem charge for cars on track owned by other railroads; and service must be competive with alternate modes of transportation. Switching strategies are also critical, not only because of the time factor, but to avoid errors and lower the expenses of engine usage within the yard. Complex decision and reporting functions are associated with car switching and train assembly. With cars being processed at rates exceeding 7,000 per day, the job has long since passed the stage where it could be done intuitively by the yardmaster.

Original Approach

The railroad company originally developed a punched card tabulator system to manage its cards. For each incoming train, a deck of cards was produced at the yard on an automatic punch controlled from the central dispatching point. Each such *advance consist* included one card for each car. When a train arrived, crewmen checked the cards against the cars and made any necessary corrections manually. The yardmaster then modeled new trains with the various card decks, using sorters and other equipment as planning aids. In a limited sense, this was a Type C-2 EDP system, with local processing of card-oriented data bases and communications in a star network centered around the dispatching office.

Deficiencies in the Original Approach

Railroad company management found this method to be inadequate for its growing needs. One problem was that manual involvement was slow and costly. Another difficulty was that the logic associated with switching and routing was too complex to be performed efficiently by hand - even with the use of tabulating equipment - particularly in view of capabilities already being offered by available computing equipment. The deficiencies of the original approach were becoming increasingly urgent, with growing pressures on the railroad for more efficient car usage and faster service.

Requirements for a New System

Railroad company officials recognized the need for a new transportation control system to supervise cars and trains as well as manage other aspects of terminal operation. They wanted a new system to:

- Supervise the location and status of all cars on the line.
- Establish and monitor car movements and associated schedules.
- Assign and distribute empty cars to satisfy projected shipper demands.
- Meet company, industry, and regulatory agency requirements for car tracing and accounting, traffic reporting, and per diem payments.
- Serve as the basis for better communication with shippers.

Alternatives

The railroad company first viewed the improvement question principally as one of centralized management, and therefore a problem of data communication. Officials decided on a Type B-3 distributed access facility, comprised of a large central computer at headquarters with communication processors at the yards. After working for some time to implement such a facility, however, company EDP specialists realized that the distributed access approach was not going to work.

One problem was that almost 99 percent of the data collected at a yard applied directly to that yard's operation; only 1 percent was actually needed at the central dispatch point. The communications facilities needed to link more than 300 remote terminals to the central computer therefore represented an unnecessarily large expense. Similarly, the capacity of the processor needed to satisfy all of the yard-oriented jobs contending for its resources was also high and therefore costly.

The company re-examined its objectives and determined that a Type C-2 semi-autonomous multiprocessor approach would better serve its needs. With such a system, decisions related to individual yards could be handled locally. Inputs (such as advance consists) could be provided automatically through the network. Likewise, reports could be transmitted to the headquarters machine as needed for overall supervisory and management information tasks.

Company officials established criteria for selecting a system which would provide these functions and help compensate for the time already lost in the initial development program. Their goals included:

- Minimization of further unanticipated development delays.

- Rapid implementation once the configuration was established because the need to replace some of the tabulating equipment was becoming increasingly critical.

- A smooth transition from the punched card approach was desirable because of the severe economic penalties which might result from a disruption.

- High availability would have to be virtually assured, since yard operation would be completely dependent on the system.

- Proven hardware and software would be sought for rapid application development and operating reliability.

- An efficient real-time executive was a prerequisite because each yard subsystem would operate in a type B-1 multiple access mode with a large number of terminals.

- An established data base management system was highly desirable for ease in tracking and manipulating the constantly changing status of rolling stock.

- The terminals would need highly engineered operator interfaces to encourage acceptance by yard personnel and to minimize input errors or misinterpretation of output messages.

- The design philosophy should include provision for establishment and enforcement of systemwide standards.

The Approach Taken

To implement its semi-autonomous multiprocessor network, the railroad company selected a system based on the Digital Equipment Corporation PDP-11 computer family. The configuration exploited Digital's standard operating software packages that take advantage of a powerful version of the BASIC programming language.

Each yard has a pair of PDP-11/45 computers (or a PDP-11/45 and a PDP-11/40) for local processing. In addition, the yard configuration includes 15 or more VT05 cathode ray tube (CRT) terminals, eight printer terminals, large main and mass memory capacity, and a pair of card reader-punch devices. The machines operate under the RSTS/E executive for terminal management. A representative configuration is shown in Figure A-7.

The dual processor approach provides the redundancy needed for high availability. The secondary system is held in a cold standby mode and is switched into service in the event of a primary processor failure. While in standby, an intermachine communication link is used to share printing, card reading, and card punching between the two machines.

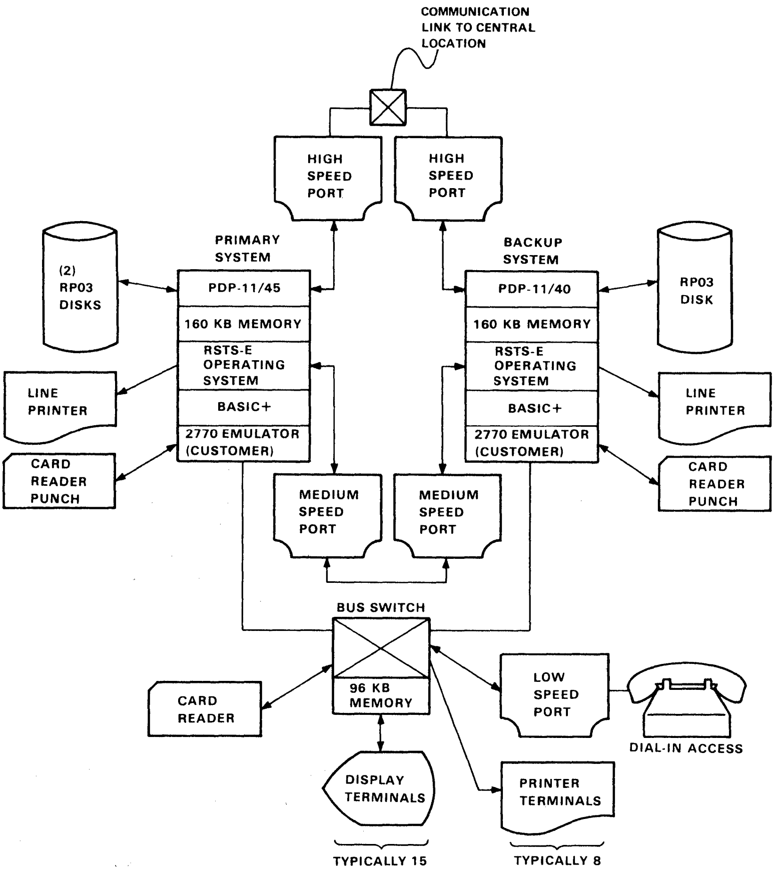


Figure A-7. A typical yard configuration.

The headquarters system is the IBM 370/168 mainframe used in the original installation. A single 2400-baud communication line is operated between each yard and the headquarters processor. The network scheme is shown in Figure A-8. Communication procedures are implemented using 2770 emulators at each yard to permit direct interaction between the 370 and the local PDP-11s.

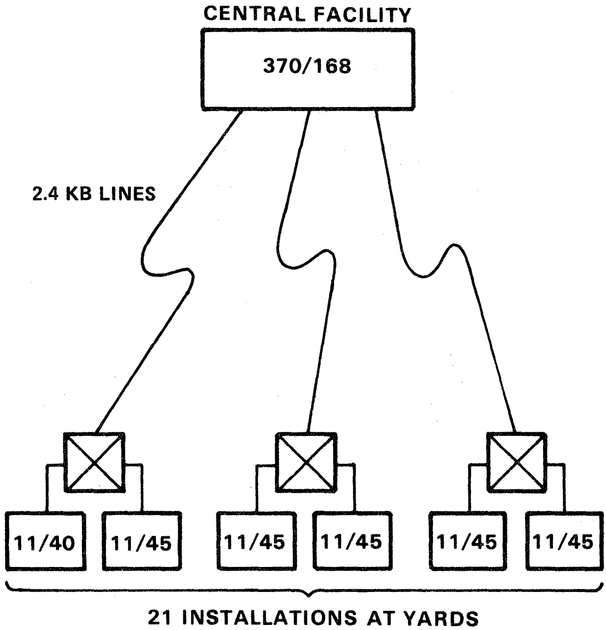


Figure A-8. Network configuration.

The first yard system was operating in 1973, and twenty additional installations have been made since that time. At present, the entire installation cycle requires about six months. The phase-over begins with the PDP-11's processing the cards (in place of the older tabulating equipment) and proceeds through increasing stages of automation for a smooth transition to a processor-to-processor network.

The capacity of the system has proved more than equal to the tasks originally contemplated, and a number of advanced applications have been implemented to take advantage of this powerful resource. Jobs now being performed include:

- Production of switch lists, including generation of preswitch analyses to permit the yardmaster to examine the effects of alternative strategies before making final decisions.

- Generation of car classification work orders.
- Maintenance of car inventories.
- Support of local management information requirements.
- Support of central reporting functions, incuding on-line inquiry capability, track and yard summaries (with aging data), Rule 15 offering reports for cars refused in interchange, car-in industry reports for stock being repaired at the yard, and various demurrage reports.

Benefits

The system has been installed and is operating at 21 locations at the same cost as the original tabulating equipment. This level of expenditure is one third to one half the cost projected for the multi-access approach.

Overall, the distributed processing system has averaged more than 98.5 percent availability since the initial installation.

The result of the system has been that throughput at the yards has increased, car handling errors have fallen, and the cost of operating switching engines has declined. The paperwork burden and the use of unit record equipment for yard office administration have been reduced. At the same time, the information available at headquarters for total operations management has increased.

The system has evolved with a uniform discipline which has proved to be easy to enforce from the central headquarters. No data processing personnel have been required at the individual terminals, and all programs are written by a central applications group and downloaded into the remote processors.

A remarkable measure of control support, not only for the system but also for operational questions, has also been achieved. This is possible because the processor at each site effectively models activity at the yard. If help is requested on switching or other tasks, members of a central control group (at headquarters) can dial in to provide detailed analyses or simulations to assist the local yard crew in making optimum decisions.

IV. AUTOMOBILE ENGINE TESTING

A major European automobile manufacturer maintains extensive research and development facilities for testing engine performance, exhaust emissions, and transmission operation. The testing is highly automated, requiring closed loop control of experimental parameters with the acquisition and recording of large numbers of data points (a representative test sequence conducted in a 64-Hz process loop yields about 1 million measurements at burst rates as high as 2,000 points/sec).

Original Approach

In originally establishing the facility, the manufacturer recognized that the desired levels of throughput and performance could be achieved only by using digital computers for test stand control and data acquisition. The company therefore installed a battery of four CDC 1700 computers, each of which operated six test stands. Two additional 1700s were employed to provide backup and graphic output capability. These six machines were supervised by a single CDC Cyber computer which stored data from all tests.

Deficiencies in the Original Approach

Although the system performed the test functions satisfactorily, the manufacturer recognized three major deficiencies. The primary shortcoming was that a test laboratory expansion was contemplated, but the present computers could not accommodate any additional stands. Moreover, the 1700s in use had become obsolete, and new units would be overly expensive to purchase even if machines were available. The second fundamental problem was that the configuration was too vulnerable to processor outages. The failure of a single 1700 would interrupt operation at six locations; a problem in the Cyber would shut down the entire site. Finally, the atmospheric and electrical environment near the test stands made it impossible to place the computers in close proximity to the equipment being controlled, necessitating installation of long runs of multiconductor shielded, armored cable. This wiring was not only expensive, but was so complex that it discouraged switching and changeover – limiting flexibility for reconfiguration as well as provision for redundant backup.

Requirements for a New System

Company officials decided to implement a new system for this application. The operational requirements would be:

- The system would have to take over the existing stations and provide a growth path for incremental expansion for 15 years – to at least 50 simultaneous test stands.

- The system would have to operate in a "fail-soft" mode, so that an outage on any machine would have no more effect than interruption of testing on a single test stand.

- The system could lose essentially no data. (All test loops can execute at 64 Hz, and at least three stands can produce 2,000 data points per second simultaneously.)

In addition to these functional requirements, in-house data processing specialists established several system-oriented criteria:

- All hardware and software would be standard to facilitate procurement, simplify development, and minimize implementation risks.

- If a multiprocessor installation were adopted, all machines would have to be compatible to permit programs to be written on a single development system and downloaded by communication links or by disk transfers to other computers. Likewise, a single communication discipline would have to be applicable at all levels in the structure.

Alternatives

The manufacturer evaluated a number of approaches to system implementation. All were hierarchical (and therefore distributed) but employed distinct concepts of assigning tasks to system elements.

The first proposal was essentially a direct replacement of the existing machines with state-of-the-art processors – having more capacity and higher speeds, but costing less than the original equipment. This implementation failed to meet the availability criterion because downtime at the supervisory level would disable the entire operation, and an outage of a control computer would put several stands out of commission. Backup could be provided to alleviate the availability difficulties somewhat, but time and data would be lost during the changeover, particularly because of the complexity of the cabling needed between the test stands and the minicomputers. This approach also presented expansion problems due to the limit on the amount of data that could be channeled into the supervisory processor at one time.

The firm considered, and rejected, an alternative two-level hierarchy. This implementation involved pairs of minicomputers – a small machine dedicated to each stand, tied directly to a second processor for supervisory control and data storage. This approach improved the expansion problem because a relatively economical small computer could be purchased for each test stand added. It also improved availability because failure of the control processor would affect only one test stand. However, complex cabling was still needed between the test stands and the first-line minicomputers, so changeover was difficult.

A third alternative was to increase stratification to three tiers by placing microprocessors directly on the test stands. This would eliminate the need for complex cabling to the control processor because simple bi-directional data lines could handle the signalling to the minicomputers. Availability would be improved because changeover would be simpler in the event of a minicomputer failure. Moreover, the microcomputers could conceivably incorporate enough logic to suspend testing and initiate a hold or orderly shutdown sequence if operation were interrupted. There was still an expansion problem, however, because of the number of lines tied to the one master processor, and a continuing likelihood of total outage if the supervisory machine were to fail.

The final configuration examined involved a four-level hierarchy. Microcomputers would be installed as input-output interfaces at the test stands. Each such controller would interact with a small dedicated minicomputer through a bi-directional data line. The minicomputers would provide control and data storage functions for individual test stands, and in turn would be clustered to a stage of minicomputers used as data concentrators. The concentrators would communicate with a central machine. This approach promised high availability because the failure of a minicomputer would affect only one test stand, and a backup machine could be switched quickly into service. The failure of a concentrator might eventually shut down a block of stands but would not interrupt work in process because all control and data acquisition associated with an individual test could be performed by the dedicated minicomputer alone. The four-level approach also allowed for expansion; the central processor could accommodate several additional concentrators, each having a multiplicity of test stand computers under its supervision.

The Approach Taken

The automobile manufacturer selected the four-level multiprocessor approach for the application with hardware, software, and communications provided by Digital Equipment Corporation. The overall configuration is shown in Figure A-9.

The processing equipment for each test stand is illustrated in Figure A-10. The control-processor is a PDP-11/34 minicomputer with two disk drives for program and data storage. The initial implementation incorporates 24 such machines, and expansion to 84 is possible.

The concentrators, each of which can meet data handling criteria with as many as 12 control computers attached, are also PDP-11/34 minicomputers (Figure A-11). These machines have no disk storage and no consoles, but do have 288K bytes of random access memory.

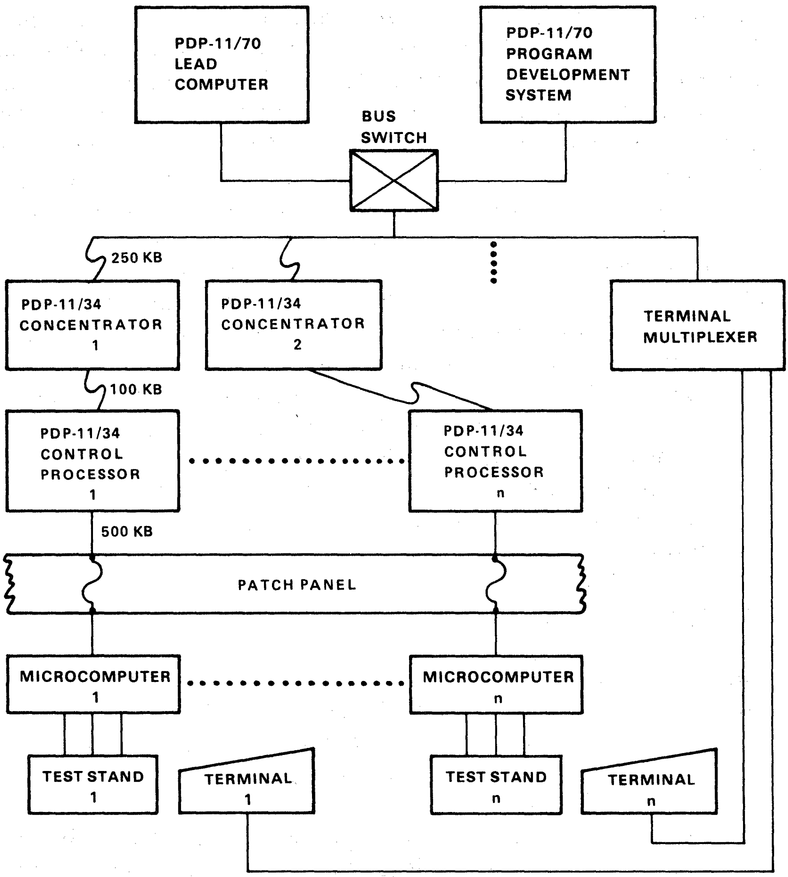


Figure A-9. Overall configuration.

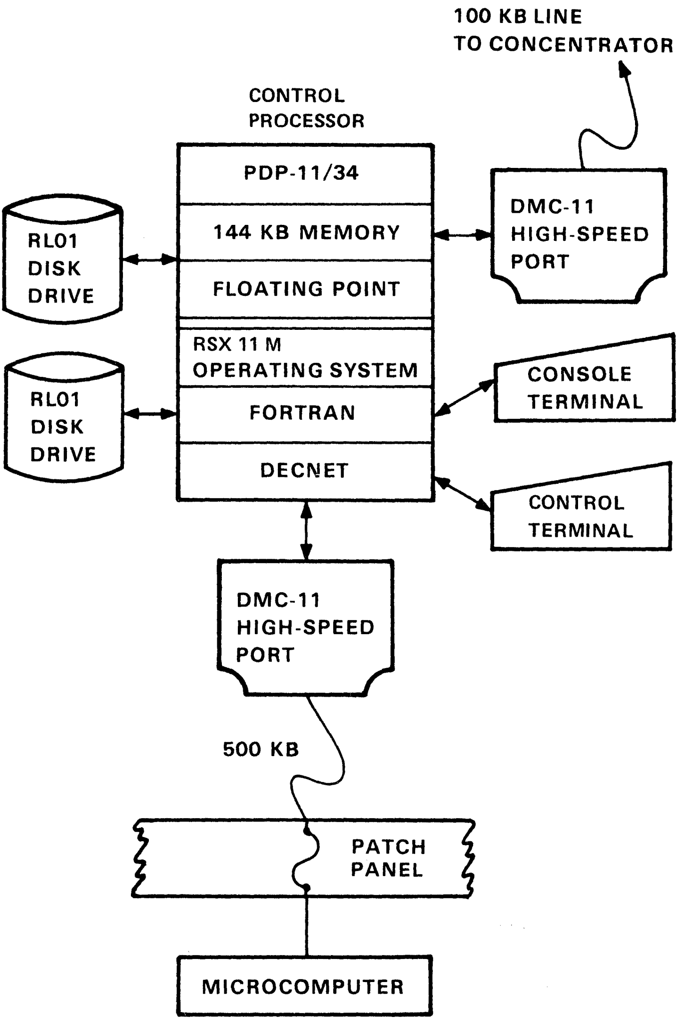


Figure A-10. Test stand configuration.

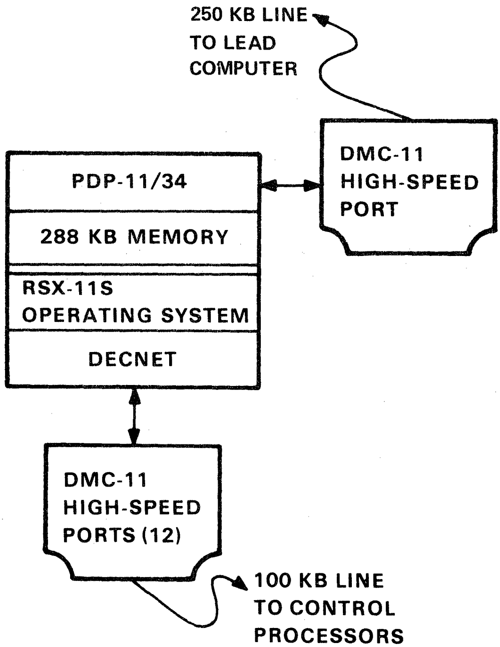


Figure A-11. Concentrator configuration.

The central computer facility is shown schematically in Figure A-12. This facility comprises two PDP-11/70 computers with a large complement of peripherals. The primary PDP-11/70 can accommodate up to seven concentrators. The secondary machine, which serves as a cold-standby for backup, is the larger of the pair and is used for program development when not on-line to the process.

Benefits

The system is able to accommodate the control and data acquisition requirements of the existing installation and offers capacity for expansion to meet current projections for the next 15 years of growth. All hardware and software is standard as required in the initial

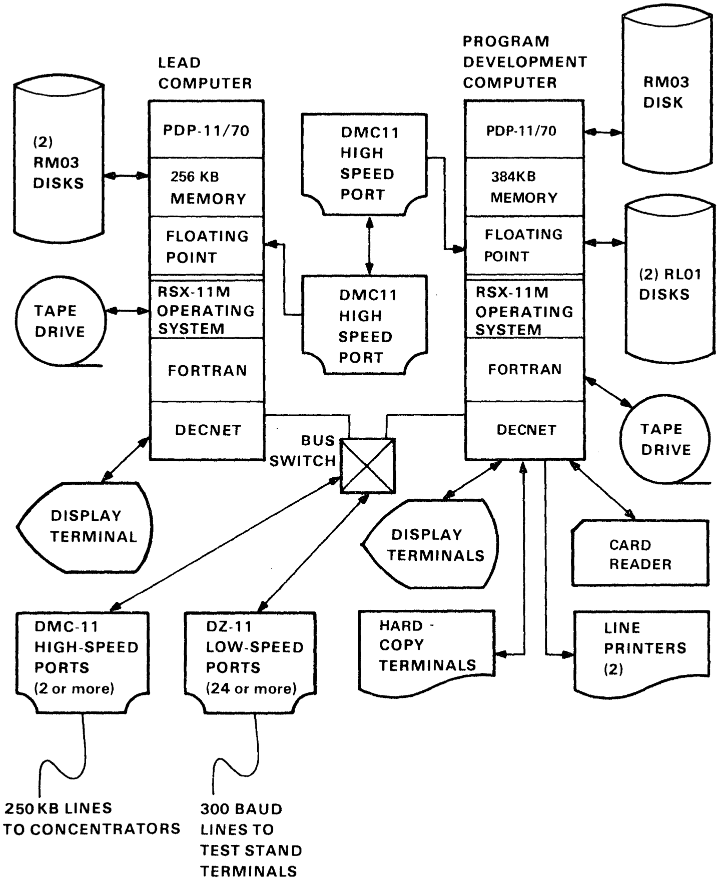


Figure A-12 Central computer facility.

specification. The communications protocol is also standard and uniform thoughout the hierarchy.

The system offers high availability. The first-level minicomputers are readily switched into operation through a patch panel within the DECnet architecture, the concentrators are invisible to test being run, and full functional backup is available to the central system.

No difficulties are being encountered because of the high data rates or the total number of points being logged. Company officials feel confident that the complexity of the tests could be increased without approaching the limits of the computer system to handle the increased load.

V. AN EMERGENCY TELEPHONE NETWORK

Emergency dialing service is being implemented by the joint efforts of telephone companies and municipal authorities throughout the country. The objective is to establish a standardized number, 911, which subscribers can reach from any telephone to contact an appropriate dispatcher.

Establishment of such a service in an existing teleplant is a complex task. For example, modifications to the signalling architecture are necessary to provide immediate dial tones at pay stations so that 911 calls can be made without depositing coins. Likewise, dedicated trunks must be implemented so that 911 calls are routed directly to the dispatchers.

Original Approach

A 911 system had been in operation in a large midwestern city for several years. The configuration was relatively conventional with all 911 calls being trunked directly from the local telephone exchange to dispatchers at police headquarters. Dispatchers had to be organized into zones corresponding to telephone exchange boundaries. The responding officer ascertained the location of the phone and type of emergency from the caller and took appropriate action.

Deficiencies in the Original Approach

Municipal authorities recognized several opportunities for improving the effectiveness of the emergency service. For example, the telephone network structure did not coincide with police or fire department boundaries. Trying to match dispatchers with telephone exchanges was an undesirable constraint. Also, officials wished to have the flexibility to shift service zones with time to follow trends in population demographics.

A second difficulty was that certain classes of callers were unable to tell the dispatcher where help was needed (because of language, age, psychological or physical state, or unfamiliarity with a neighborhood). This prevented reaction to certain situations and delayed the response to others. A related problem was that some callers simply would not provide the necessary identification, making it difficult for the dispatcher to distinguish between crank calls or false alarms and anonymous good Samaritans.

Requirements for a New System

Telephone company and municipal authorities recognized that electronic data processing (EDP) equipment could be interfaced with the 911 system to provide an enhanced level of service. They developed the following set of functional requirements for a new system:

- Selective routing of calls to dispatchers was needed so that telephone, police, and fire boundaries could be established independently (and changed when desired) with calls trunked to the proper dispatcher.

- Automatic Location Identification (ALI) would solve the problem of establishing the source of calls when verbal communication is impossible and would also trace false alarms.

- Means should be available to facilitate dispatch to responsible agencies: fire, medical, or police.

The authorities determined at the outset that the functional requirements were technically feasible. Computing equipment had been interfaced with the telephone network at the signalling level in many applications. For example, computers were being used in number identification systems for toll billing and traffic monitoring. They accordingly established the following criteria for evaluating EDP approaches that could be used for the 911 task.

- Compatibility with the existing telephone network, including not only the signalling structure, but also the lines already installed

- Extremely high availability
- Ability to handle a total data base comprising more than 2.6 million subscriber locations and 10,000 street address and emergency zone correspondences
- Capacity to process at least 40,000 calls per day, with up to 5,600 calls during any one busy hour, and to provide response within one second
- Restriction of access to the telephone company master directory (to safeguard privacy).

Alternatives

The first approach considered involved a centralized computer managing the data base and providing the commands necessary to operate the switches in the routing network. Such a system would, of necessity, be quite large in order to perform all the required functions. Cost would therefore be high, particularly if redundancy were implemented to insure high availability. To protect the data base from unauthorized access, the facility would have to be located at a central telephone company office and would require replacing vast amounts of cable already run to switching equipment in the police department. There was also skepticism about the ability of a single system to handle the required call volume without severely degraded response during peak periods.

A distributed multiprocessor configuration was studied as an alternative. Such a system would meet the requirements for compatibility with telephone signalling protocols and could be implemented with processors at telephone, police, and fire offices using existing cable installations. In a system based on small processors, redundancy would be more economically achieved and would therefore be a cost-effective means of insuring the required high availability. Accommodation of a large data base, with assurances of strict security enforcement, was another advantage of this approach. This was possible because the overall file structure could be segmented naturally into a subscriber directory and zone correspondence tables for the fire and police departments. Finally, the speed criterion was relatively easy to meet because processor capabilities could be matched to each part of the application.

The Approach Taken

The system chosen for the application is a semi-autonomous distributed processor configuration. Minicomputers are located at telephone company and various police and fire department locations. The network is connected in a star configuration (Figure A-13). The data base for the location directory is at the hub under the control of the telephone company, and switching processors are located at police and fire department dispatch centers. All processors are members of the Digital Equipment Corporation PDP-11 family. All inter-processor communications are implemented with DECnet protocols using 4800-baud synchronous communication lines.

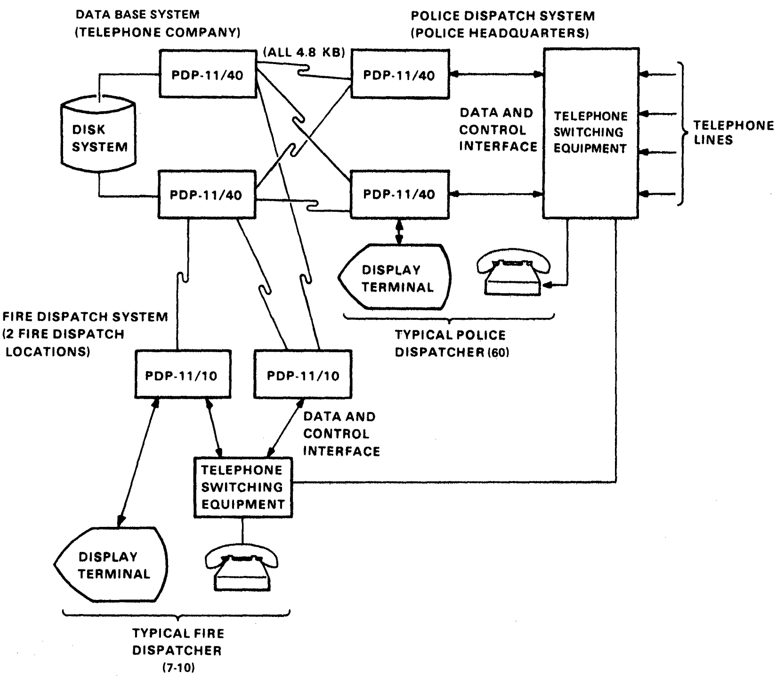


Figure A-13. 911 communication network.

The telephone company installation, shown schematically in Figure A-14, acts as a directory manager. A pair of PDP-11/40 processors is used with four dual-ported disk drives. This is arranged as an overflow backup system which insures that one system will be available when the other is busy, overloaded, or out of service. Each machine has a complete data base, so two copies are on-line simultaneously.

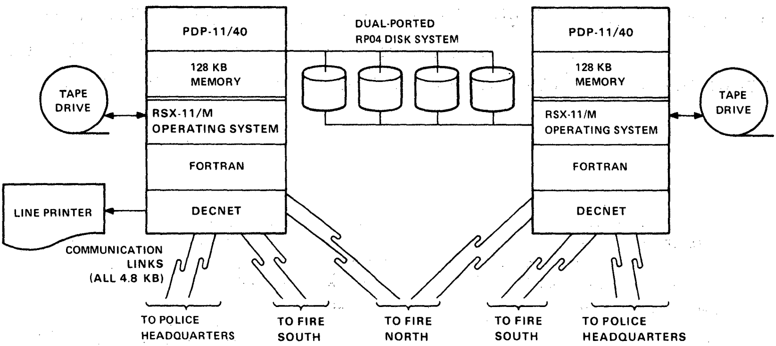


Figure A-14. Data base system.

The police system comprises a pair of PDP-11/40s operated in a watchdog hot-standby mode (Figure A-15). Dispatch terminals are connected to the system through a special switch to provide redundancy for immediate changeover, but without the confusion that would result if different addresses were displayed on different terminals. Smaller PDP-11/10 processors, in a configuration similar to that used by the police, are employed by the two fire department dispatch centers.

When a 911 call is recognized at a local exchange, it is switched through the telephone network to the switching equipment at police headquarters along with the identification of the originating number. The 11/40 at the police center sends a request to the central data base for the corresponding address and zone. The information

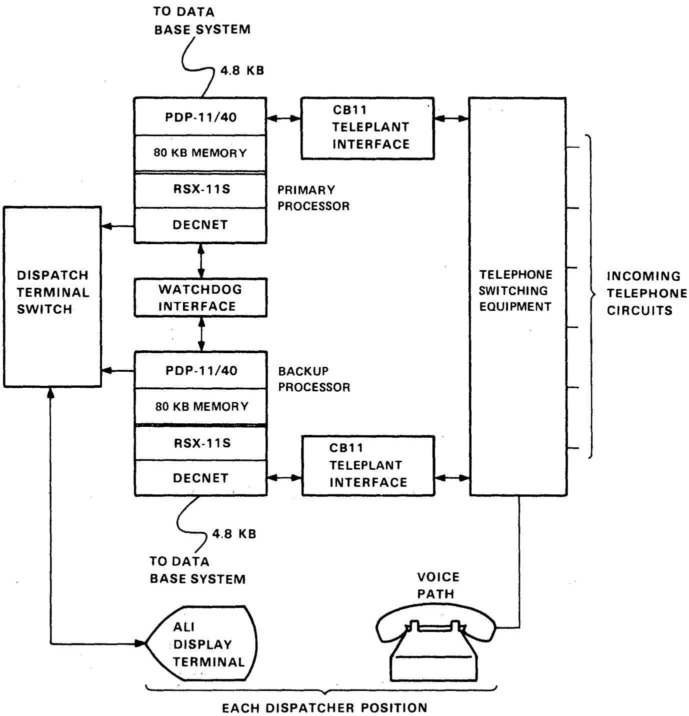


Figure A-15. Police dispatch system.

is extracted from the master directory and returned to the police system. The zone identification is used to determine routing to the proper dispatcher. When the dispatcher answers the call, the telephone number and address are displayed on the ALI display. If the call is a police problem, the dispatcher makes the appropriate disposition. If the caller indicates that the problem is a fire, the dispatcher depresses a single button on the console which transfers the call, including the number and address, to the appropriate fire department processor. The call is routed to the designated dispatcher in that agency in a manner analogous to the initial transfer. Voice communication is unnecessary to establish location. Moreover, the address display can be held by the system after the phone is hung up at the point of origin.

Benefits

The operational availability of the computer system has exceeded 99.95 percent. The system can handle 40,000 calls daily, with up to 5,600 calls in any one hour. It is interesting to note that these limitations are imposed by the teleplant rather than the computer system.

The data base has room for 4 million subscribers, with only 2.6 million needed at present, and can accommodate 10,000 street-zone cross-references. To date, the system has maintained a response time of within one-quarter of a second - measured from the moment the police system is notified of an incoming 911 call to the switching of the teleplant crossbar. This fast response has not only been credited with great improvements in the ability of the agencies to cope with emergencies, but, coupled with automatic location identification, has led to the apprehension of fraudulent callers and reduced the instance of false alarm.

The system was installed without any major changes in the line structure of the existing 911 network, and therefore represents considerable economy relative to the other configuration considered. The system also offers the benefit that each data base is under the control of the appropriate agency, thereby maintaining the security requirements of the telephone company, and accommodating establishment and shifting of boundaries by the police and fire departments.

VI. MINI CASE STUDIES

A. A Wire Service Copy System

A major international wire service is using distributed processing for copy entry, editing, storage, and transmission in many cities throughout the United States and abroad. At each site, the wire service has installed a copy editing system provided by a Digital Equipment Corporation OEM. These systems use PDP-11 processors for story editing and terminal interface and use PDP-8E processors for data base management.

A typical installation in one foreign country makes use of a two-level distributed processing system. In small cities, video display equipment is used to enter stories; it also provides limited editing and storage capabilities. At the wire service headquarters in the country, large systems are used to store copy in storage queues from which they can be recalled for review and further editing. The headquarters location also contains directories for all stories in the queue.

Clerical personnel enter stories at either the remote or headquarters location, using upper and lower case video terminals (up to eight at each location). Any amount of copy – from a single character to entire paragraphs – can be altered, inserted, or removed. Once the stories have been entered, they are transmitted from the remote city to the headquarter. They can also be printed out at this time for local review.

At headquarters, copy is stored in appropriate queues (sports, financial, news) until it is edited and dispatched over the news wires.

The electronic news management system was adopted because of its ease of news copy editing and retrieval at any stage of preparation, the ease of queuing stories for rapid handling, and the capability for quick transmission of messages and copy. The news service believes that its implementation has resulted in significant material and man-hour savings through almost total elimination of paper shuffling and redundant typing of copy.

B. Food Distribution

A major food manufacturer is using distributed processing to provide a centralized system for order entry and inventory update. The manufacturer has four regional order entry facilities located in the middle Atlantic states, the Midwest, the Southeast, and the West Coast. At each of these sites, the food manufacturer has installed a distribution management system provided by a Digital Equipment Corporation OEM. These systems are based on PDP-8 or PDP-11 processors and include video displays, printers, and disk storage as appropriate to the size of the distribution center.

Salesmen telephone their orders each evening to the appropriate regional office where they are entered into the system using the video display terminals. The system acknowledges each order received. Acknowledgments for valid orders, whether rush orders or regular orders, are printed on special acknowledgment forms; suspended orders are acknowledged on plain paper. Orders which have not been accepted as billable by the system because of some irregularity are printed as "invalid". Order correction and order splitting procedures are used to correct invalid orders to make them shippable.

Bills of lading are generated at convenient intervals during the day, and the carrier summaries are generated at the end of the day. These are either transmitted directly or mailed to the appropriate distribution warehouses. In addition, the regional facilities generate shipping documents for rush orders.

At the end of each day, all work orders entered during that day are transmitted to a central computer facility via a communications network. This is done after normal working hours to minimize communications costs.

The regional facility also receives from the central facility bills of lading and carrier summaries for orders which have been processed earlier. These are printed and mailed to the warehouses from the regional facility during the night.

The food manufacturer sees two major benefits from the installation of the distributed processing system. The first benefit is a reduction in inventory made possible by the regionalization of order processing and control. The second major benefit is a significant improvement in the accounts receivable of the company.

C. Agency Accounting for a Railroad

A major railroad has installed a distributed processing system to control its revenue accounting. The agency accounting system has four major applications: rating, waybilling, accounts receivable, and miscellaneous accounting. The distributed processing system consists of PDP-11/70 computers in seven major cities, with terminal equipment located in 68 cities. A central computer houses all remote programs and master files which it distributes to each remote computer as appropriate. Each remote PDP-11/70 services satellite input/output terminals which are strategically located to support the freight agencies of the railroad.

Railroad rates are complex and difficult to maintain and apply manually. The rate data base requires 60,000 rate records covering 23,000 unique forwarded moves. The frequent changes in the rates makes it difficult to maintain accurate and up-to-date tariff manuals at the remote locations where rates are applied. The on-line rating system is a mechanized application of the rate during preparation of revenue waybills. This can be generated either directly by the computer without assistance from rate personnel or by interaction between the computer and rate personnel. Approximately 70 percent of the railroad's originating traffic can be covered by repetitive rates from the computer data base.

The outbound waybilling application captures available data at the time and place of initial transactions and adds data at the time and place of additional actions. It then organizes and maintains these records in a central location accessible for all uses.

The railroad sees the following advantages to the use of the agency accounting system:

1. It eliminates costly duplication of records and insures that all departments base decisions on the same data.

2. It achieves and maintains accurate rating functions.

3. It maximizes productivity of rating personnel.

4. It improves customer payment performance by providing efficient freight billing.

5. It provides a sufficient degree of operational independence in the event of a central outage.

6. It permits decentralization of division functions but maintains central controls.

7. It provides access to information both centrally and locally.

D. On-Line Pharmacy System

A major pharmacy chain is using distributed processing to provide an on-line system for billing and patient profiles. In the initial installation of the system, terminals are being installed at 30 pharmacies in a single area. Plans call for dissemination of the system throughout the country.

The system uses microprocessors to control the CRT terminals located in each store. The microprocessors are connected by a multipoint DECnet protocol to twin PDP-11/70 processors located at headquarters.

The system provides the pharmacist with the following capabilities:

- inventory control over all inventory in the store
- labeling of prescriptions
- patient profiles
- drug inquiry
- drug pricing
- drug interaction

The pharmacy chain sees the principal benefit of the system as providing computer facilities to the individual pharmacy without installing a computer system at each pharmacy. The use of distributed processing makes it possible to satisfy many of the processing requirements on the microcomputer at the pharmacy, using the headquarters computer for central data base storage and retrieval.

E. Telephone Network Performance Measurement System

A major international telephone network is planning to use distributed processing to provide an integrated performance measurement system for its network operations. The network covers a large geographical area and is divided into autonomous regions, each with a high degree of local control. Approximately 90 percent of all telephone calls originating in any region terminate in that region; only 10 percent of the calls move from one region to another, although these are typically the highest cost calls. The telephone network wishes to obtain a clear continuing measure of network performance in order to match the performance of the network against management goals and customer expectations.

In considering its approach to network performance measurement, the telephone network considered both the centralized and the distributed processing approaches. It eventually chose the distributed processing approach for two major reasons:

- The fact that 90 percent of the calls originating in a region terminate in that region suggested that communication costs would be high if all collected data were sent to a central location for processing.

- The fact that the regions are largely autonomous and quite concerned that they have the principal control over their data suggested that the data processing be regionalized.

The approach the telephone network is planning consists of placing one processing facility in each region plus one additional facility at a central location. All locations will be connected to provide data from one region to another.

Because the regions vary (quite widely) in size, it is advantageous to be able to use a compatible family of hardware and software to span the various regional processing requirements. For this reason, the telephone network is considering the use of the PDP-11 family which permits small processors and disk storage capacity to be used in the smaller regions. Large processors and disks can be used in the larger regions, all in the context of a consistent operating system environment. It is planned that DECnet will be used as the interprocessor communications facility.

The telephone network expects two major benefits from the installation of this system:

- It will provide the ability to measure network performance in a manner consistent with customer expectations.

- It will provide management at all levels in the telephone company with clear, unambiguous measures of performance of the resources under their control. In this way, the telephone network hopes to substantially improve its operation.

F. Marketing Distribution in an Oil Company

A major oil company is using distributed processing to provide an integrated system for physical distribution. Its analysis indicated that physical distribution was its third largest cost of doing business (following manufacturing and marketing).

At each distribution center, a minicomputer-based system keeps all the data files and provides interactive application programs to handle five separate functions of physical distribution:

- transportation of raw materials and finished products into and out of the distribution center

- material handling within the plants and warehouses

- inventory control to assure optimum levels of stock

- order processing of all documents (sales orders, invoices, bills of lading, payments)

- communications both within the distribution center and between it and central management

Customers call the distribution center at any time during the day to place orders. The orders are entered through interactive terminals by an order entry operator. Inventory files are kept current to the minute. Customers are advised immediately whether adequate stock is available and are queried as to what action to take if an out-of-stock condition exists.

The system then produces picking tickets for the warehouse and loading information for trucks to deliver items to the customer. The system also generates invoices and keeps track of receipts. At the end of each day, all information is sent to a headquarters system which keeps track of the company's overall inventory on a batch basis.

The system is based on a PDP-8A computer with a large disk storage system for the inventory files. Typically, ten interactive CRT terminals are used for data input and inquiry, and four to six print-only terminals are used to produce pick lists, invoices, etc. A magnetic tape drive is used to back up the disk files. Finally, a high speed port is used to support communications to the headquarters facilities.

The company sees benefits from the system in the following ways:

- white collar labor savings due to large reductions in the clerical work order processing, invoicing, inventory tracking, loading, and shipping control

- blue collar labor savings due to more efficient warehouse operations, vehicle loading procedures, and scheduling of replenishment.

- reductions in inventory levels due to enhanced inventory control procedures

- reduced EDP costs in the data center

- reduced level of accounts receivable due to faster order processing

- fewer lost sales due to having fewer out-of-stock conditions

- more effective use of truck fleets

Appendix B
DIGITAL EQUIPMENT CORPORATION PRODUCTS

Many of the 100,000 Digital Equipment Corporation minicomputers sold in the last two decades have been used in distributed systems. Traditionally, minicomputer systems have been interactive and have focussed on a specific function or application. Such systems are easily adapted to be components in a distributed computer system.

This chapter presents a cursory overview to some DIGITAL hardware and software products that can be used to form distributed systems. The chapter is not a catalog, however: many products have been omitted entirely; only partial details are given for those described. A local DIGITAL Sales Office can provide full product details, prices, and delivery information.

COMPUTER SYSTEMS

DIGITAL produces computer systems that range from small to upper-middle size. There are three major system families: the PDP-8, the PDP-11/VAX-11, and the DECSYSTEM-10/20. Each family is characterized by software and system compatibility.

The PDP-8 Family

By today's standards, the PDP-8 is a very simple computer. The PDP-8 instruction set is small, and the machine uses 12-bit words. The first PDP-8 was shipped in 1965. Since that time, over 40,000 machines have been produced to establish the PDP-8 as one of the most sucessful computer designs ever.

The machine was designed for simple laboratory, process control, and communications functions, but has ended up being used for countless unanticipated applications. For example, this book was composed on a PDP-8 based word processing system, and typeset with another PDP-8.

The PDP-8 is actively marketed in two models today:

- **The PDP-8/A.** The PDP-8/A is configured as a traditional minicomputer – a box with circuit boards for processor, memory, and perhipheral controllers. Up to 128K words of 12-bit memory can be added. A fast floating-point arithmetic processor is an option. Peripherals include various disk drives, tapes, printers, paper tape equipment, instrumentation adapters, and communications controllers. The PDP-8/A can be used as a conventional computer system, or incorporated as a component in other equipment. DIGITAL sells the PDP-8/A as a word processor (WPS-102 and 200 series) and as a small business system (DEC Datasystem 310).

- **The VT-78 Video Data Processor.** The VT-78 is a complete computer system within a video display terminal. The VT-78 uses a single IC PDP-8 processor, 16K words of memory, and includes built-in controllers for floppy disks, communication lines, and a printer. The VT-78 can be used as a small general purpose system, and is also available configured as a word processor (WT-78) and a small business system (DEC Datasystem 308).

The PDP-11, VAX-11 Family

The PDP-11 has been a leader in establishing the broad applicability of the minicomputer. The first model, the PDP-11/20, was produced in 1970. Since that time, many different models have been produced: presently they span a range from the LSI-11/2, which has a complete processor on a single, small circuit board, to the VAX-11/780, which is comparable in all respects (except price) to mid-sized mainframe systems.

The PDP-11/20 was a very sophisticated small computer, with a rich logical instruction set, capable of addressing up to 28K words of 16-bit memory. Enhancements to the design since have added

- special-purpose instructions for commercial and scientific processing
- very fast computational processor options
- memory mapping; the 11/70 can address up to 2 million words of memory

The VAX-11/780 represents a major enhancement to the design, and would consitute an entirely new computer family except for the fact the VAX-11 systems are capable of executing PDP-11 programs (as well as VAX-11 "native" programs), and utilize existing PDP-11 peripheral devices. The VAX-11 utilizes a virtual memory design: a VAX-11 program can address over 4 giga (10^9) bytes of address space, whereas a PDP-11 program is limited to 64K bytes.

The following PDP-11 models are only part of the total product offering. They are presented to give some idea of the breadth available.

- **The LSI-11.** The LSI-11 is based on a custom-designed integrated circuit processor. The LSI-11 is sold as a component to be integrated into equipment (individual circuit boards), and as a complete small computer system – the PDP-11/03. The LSI-11 can have up to 56K bytes of memory, has microcode commercial and scientific instruction options, and can be equipped with a variety of peripheral options.

- **The PDP-11/34.** The PDP-11/34 is designed to be a small, complete computer system. Up to 256K bytes of memory can be used; memory mapping is standard. Instruction options are available, as is an optional memory cache. The PDP-11/34 can be used with any of the PDP-11 Unibus-attached peripheral options.

- **The PDP-11/70.** The PDP-11/70 has a built-in memory cache and map, and can have up to 4 million bytes of memory. Instruction options are available. PDP-11/70 systems can use any of the Unibus or Massbus peripherals.

- **The VAX-11/780.** The VAX-11/780 features a new virtual memory design, a revised instruction set with commercial, scientific, and special operating system and language instructions as standard instructions. The VAX-11/780 can utilize all Unibus and Massbus peripherals, and was designed for up to 8 million bytes of memory.

The PDP-11 family has a complete line of computer peripherals (so complete as to require its own handbook). PDP-11s have been used to implement a range of applications too broad to be easily categorized. On the order of 50,000 PDP-11s have been sold so far.

The DECSYSTEM-10/20

The DECSYSTEM-10/20 family of computers was first introduced in 1964 as the PDP-6 – a 36-bit machine designed for timesharing. A second model, the KA-10, began the DECsystem-10 family in 1967. In 1976, a fourth processor model, the KL, was produced both as a DECsystem-10, and as the beginning of a new family, with a new operating system, the DECSYSTEM-20. In 1978 a low-cost version of the -20, the DECSYSTEM-2020, was introduced. All told, about 1,000 of these systems have been installed.

Presently, the DECSYSTEM-20 is marketed in two classes of systems:

- **The DECSYSTEM-2020.** This low-cost version of the -20 is designed to extend the -20 family into the traditional minicomputer price range. The machine follows traditional minicomputer design by using normal line-current power and not requiring air conditioning. Up to 2 million bytes of memory can be added. The -2020 uses PDP-11 Unibus peripherals.

- **The DECSYSTEM-2040, -2050, -2060.** These are the more powerful versions of the -20, and are designed in the style of a mainframe computer. More memory can be used, and controllers have been built-in for the high-speed Massbus peripherals.

SOFTWARE SYSTEMS

In the past, the knowledgeable computer consumer spoke in terms of model numbers and microseconds. Today many systems are purchased as complete hardware/software systems, or even just as system machines. The reason for this change in focus is the rapid decline in the cost of hardware, and the relatively constant cost of software. In terms of the total cost of ownership, it's a lot more important to have the right software (to minimize programming costs) than it is to have the last iota in hardware performance.

Some of the major DIGITAL software systems are described below, categorized by typical application use.

Data Acquisition and Process Control

These systems are used for data collection process control. They are characterized by:

1. **Internal efficiency:** The real time nature of the applications demands that the system be able to respond quickly to an external event or request.

2. **General design:** Because of the diverse nature of these applications, the system design must support many styles of application.

3. **Broad device support:** The software system should include the control programs for many input/output devices.

4. **High-level language and data management support:** The system should support the commonly used high-level languages, and provide data storage and retrieval mechanisms.

These operating systems are often used as the basis for other kinds of applications, such as communications processing (e.g., message switching) or high-performance data processing (e.g., credit card validation) because of the rapid-response, high-performance focus of the designs.

The following software products are examples of DIGITAL's real-time operating systems.

- **RT-11.** RT-11 is a software system designed for relatively simple, high-performance applications on the lower-range PDP-11 processors. There is a minimum of system-introduced overhead. The system is designed to be easily usable. In addition to the basic operating system facilities (including a complete data file system), FORTRAN IV, APL, and BASIC are available. DIGITAL also markets RT-11 tailored for specific commercial and laboratory applications. The commercial version includes an interactive commercial programming language, DIBOL.

- **RSX-11M.** RSX-11M is designed for more complex, real-time applications implemented on middle-range to top-end PDP-11 systems. RSX-11M permits complex multiprogramming, provides preprogrammed support for a wide variety of devices, supports many programming languages (including FORTRAN IV PLUS, BASIC PLUS 2, and COBOL), has a full functionality file system (FILES-11), indexed access methods for record management (RMS-11), and a CODASYL compatible database management system (DBMS-11). DIGITAL also markets RSX-11M tailored for specific applications, such as plant management.

Transaction Processing

Transaction processing is the style of interactive, transaction-oriented processing that has been featured in this book. Transaction applications *can* be created without any special purpose software. A transaction processing system provides many valuable functions, such as the mechanisms needed for high data integrity, in a pre-programmed form; this greatly reduces application development costs.

- **TRAX-11.** TRAX-11 is a PDP-11 software system designed specifically for transaction processing on the PDP-11/34 and -11/70. The system is designed for forms-based interaction using an intelligent terminal – the VT-62. Transaction programs can be written in COBOL or BASIC. FILES-11 file management and RMS-11 record management services are provided. Data integrity functions and utilities are provided. The system design is optimized for transaction processing and data access.

- **RSTS/E and TOPS-20.** DIGITAL also markets transaction processing packages for these general-purpose systems. These packages reduce the development costs of transaction-oriented applications.

Word Processing

DIGITAL markets a family of word processing systems (WPS). These systems have the following features:

- A video display terminal that gives the system user a pictorial view of the text while it is created, edited, and formatted.

- A function-button editing language that gives the system user a powerful but simple-to-use set of functions for text editing, rearranging, and formatting.

- A disk unit capable of storing many documents.

- An optional printer that produces high-quality output, including justified margins, proportional spacing, multiple columns, etc.

The DIGITAL WPS systems feature word processing integrated with data processing. The systems can be used as small business systems, as well as for word processing, and can be integrated into large distributed information systems by use of communications options.

General Purpose Systems

DIGITAL sells a family of general purpose, interactive timesharing systems. These systems can be used concurrently by many terminal users, each developing, testing, and executing programs. These systems can also be tailored for specific processing applications.

- **RSTS/E.** RSTS/E is an operating system for the PDP-11/34 and -11/70. RSTS began as a BASIC programming system, but has evolved to provide multiple language programming in FORTRAN, APL, RPG II, and COBOL. RMS record management services are available.

- **VAX/VMS.** VAX/VMS is designed to utilize the extended capabilities of the VAX-11/780. It provides demand paging, working set management, and swapping as virtual memory services. The native machine can be programmed in FORTRAN, BASIC, and COBOL. Additionally, many PDP-11 utilities and languages can be used under the compatability mode feature. There is a full file-management system with record management utilities.

- **TOPS-20.** TOPS-20 is the operating system for the DECSYSTEM-20. Like VAX/VMS, TOPS-20 is a virtual memory operating system. COBOL, FORTRAN, BASIC, APL, CPL, and ALGOL are available as programming languages. A CODASYL Database management system – DBMS-20 – is also available.

Communications Software

The interactive systems discussed above each supports some set of asynchronous and synchronous terminal devices. DIGITAL manufactures a wide variety of communication peripherals. The DECnet interconnect software was described in Chapter 10.

INDEX

DISTRIBUTED SYSTEMS HANDBOOK READER'S COMMENTS

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our handbooks.

What is you general reaction to this handbook? (format, accuracy, completeness, organization, etc.) ______________________________

____

____

____

What features are most useful? ______________________________

____

____

____

Does the publication satisfy your needs? ______________________________

____

____

____

What errors or omissions have you found? ______________________________

____

____

____

Additional comments ______________________________

____

____

Name ______________________________

Company ______________________________ Dept. ______________

Title ______________________________

Street ______________________________

City ______________ State ______________ Zip ______________

(staple here)

(please fold here)

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY
IF MAILED IN THE UNITED STATES

Postage will be paid by:

DIGITAL EQUIPMENT CORPORATION
SOFTWARE ENGINEERING
ML12/A62
146 MAIN STREET
MAYNARD, MASS. 01754

Re: Distributed Systems Handbook

(staple here)

"Digital Equipment Corporation is proud to have been part, along with our customers, of many of the most innovative and foresighted uses of computers. From the beginning we have believed that computers should be tools that can be used by people who need information to do their jobs. We have promoted the design of interactive computer systems that can be placed where they are needed. We see the trend toward the increased use of interactive, distributed computer systems as confirmation of our basic philosophy."

Kenneth H. Olsen
President

digital

PRINTED IN USA EB PI195 03/78 08A 02 20