John Kinnear

Report Series
No 18

Distributed Processing: Management Issues

April 1980

The Butler Cox Foundation

The Butler Cox Foundation

The Butler Cox Foundation is a research group that examines major developments in the fields of computers, telecommunications and office automation on behalf of its subscribing members. The Foundation provides a set of 'eyes and ears' on the world for the systems departments of some of Europe's largest organisations.

The Foundation collects its information through its office in London and also through its associated offices in Europe and the US. It transmits its findings to its members in three main ways:

— Through regular written reports that give detailed findings and substantiating evidence.

— Through management conferences for management services directors and their senior colleagues, where the emphasis is on the policy implications of the subjects studied.

— Through professional and technical seminars where the members' own specialist managers and technicians meet with the Foundation research teams to review their findings in depth.

The Foundation is controlled by a Management Board whose members include representatives from the Foundation member organisations. The responsibilities of the Management Board include selecting topics for research and approving the Foundation's annual report and accounts, which show how the subscribed research funds have been employed.

The Management Board

David Butler	Chairman, Butler Cox & Partners Limited, Morley House, 26-30 Holborn Viaduct, London EC1A 2BP Tel: 01-583 9381
Michael Colin	Director of Advanced Systems, BL Systems Limited, CDC, PO Box 5, Redditch, Worcestershire B98 9HB Tel: (0527) 60133
George Cox	Managing Director, Butler Cox & Partners Limited (as above)
Martin Ray	Director of the Butler Cox Foundation, Butler Cox & Partners Limited (as above)
Bill Sherratt	Information and Computing Services Division, Shell UK Limited, PO Box 148, Shell-Mex House, Strand, London WC2R 0DX Tel: 01-438 3000
Bob Tester	BOC Limited, Hammersmith House, Hammersmith Bridge Road, London W6 9DX Tel: 01-748 2020

Publications 1977-80

Reports 1977-80

1/July 1977	Developments in data networks
2/August 1977	Display word processors
3/November 1977	Terminal compatibility
4/December 1977	Trends in office automation technologies
5/February 1978	The convergence of technologies
6/April 1978	Viewdata *(Out of print)*
7/May 1978	Public data services
8/June 1978	Project management
9/July 1978	The selection of computerised PABXs
10/November 1978	Public on-line information retrieval services
11/February 1979	Improving systems' productivity
12/June 1979	Trends in database management systems
13/April 1980	Cost trends in data processing
14/September 1979	The competitive equipment market
15/September 1979	Management services and the microprocessor
16/November 1979	Mainframe computers in the 1980s
17/February 1980	Electronic mail
18/April 1980	Distributed processing: management issues
19/June 1980	Office systems strategy
20/August 1980	The interface between people and equipment
21/October 1980	Private communications networks

Conference Transcripts 1977-80

1/May 1977 London	Computers, communications and office automation: the emerging pattern of information systems
2/November 1977 Stratford-upon-Avon	The management policy issues arising from emerging technologies
3/April 1978 London	Key issues in data processing
4/October 1978 Bristol	Management priorities for the 1980s
5/May 1979 Westchester, New York	Office automation: when and how will it happen? Distributed processing: the policy issues for management services
6/November 1979 Birmingham	The changing role of management services
7/May 1980 Venice	System development — prospects for the 1980s
8/December 1980	New applications, new technologies

SISDOCONSULT

THE BUTLER COX FOUNDATION

Seminario Tecnico Professionale

2 Ottobre 1980

INFORMATICA DISTRIBUITA: PROBLEMI DI GESTIONE E DI DIREZIONE

PROGRAMMA DELLA GIORNATA

9.00 ÷ 9.25	Registrazione e caffè	
9.25 ÷ 9.30	Introduzione e presentazione	A. Oskian Sisdo Consult
9.30 ÷ 10.15	Aspetti strategici e politici della elaborazione distribuita	J. Kinnear Butler Cox & Partners
10.15 ÷ 10.35	Intervallo	
10.35 ÷ 11.35	Strategie organizzative e Informatica Distribuita	P.C. Maggiolini Univ. della Calabria
11.35 ÷ 12.30	Le Unità Sanitarie Locali come problema tipico di informatica distribuita	F. Tucci Regione Basilicata
12.30 ÷ 14.00	Colazione di lavoro	
14.00 ÷ 15.15	Il disegno delle reti nei sistemi distribuiti	P.S. Olivotto Digital Equipment Italia
15.15 ÷ 15.35	Intervallo	
15.35 ÷ 16.30	Personalizzazione di sistemi predefiniti o costruzione di sistemi su misura ?	L. Floridia RAI-Radiotelevisione It.
16.30 ÷ 17.15	Discussione e conclusioni	R. Bellini Sisdo Consult

La relazione del Sig. J. Kinnear sarà basata sul rapporto di ricerca n° 18.

Milano - Hotel Michelangelo, Via Scarlatti 33

Abstract

Report Series No 18

Distributed Processing: Management Issues

by John Kinnear

April 1980

Almost since the earliest days of commercial computing, most computer applications have been designed, operated and, to a large extent, controlled by data processing professionals. Recently, there has been a move towards decentralising computer systems. This represents a radical departure from previous practice, in that it places the control of the computer system in the hands of the end-user. Bearing in mind that data processing departments often took away the end-user's control of his system in the first place, it is perhaps surprising that anyone should object to this reversal of an established trend. However, many hard lessons have been learned about the practice of computing over the past decade or two, and many data processing people are genuinely concerned that decentralised processing will eventually lead to total chaos in corporate systems. What we define as distributed processing lies midway between the fully decentralised and the traditional centralised approach, because it seeks to ally the benefits of decentralised equipment with a degree of overall co-ordination and control.

Distributed computing (which embraces both decentralised and distributed processing forms) is not a new trend — the earliest systems of this type were implemented in the early 1970s. It has been discussed extensively over the past half dozen years or so, and has been extolled and denigrated by turns. There is now enough hard evidence to show that it is a genuine change in the direction of data processing and it is, in our view, a change that will endure.

Distributed computing is not only a different way of implementing computer systems. It also has a number of implications for the way that organisations shape their computing policy, and for the role both of management services and the data processing department. In particular, it raises two key questions:

- How should the move to distributed computing be controlled and directed?

- How much responsibility should be devolved to end-user managers?

This report discusses the pros and cons of distributed computing. It describes user experience and the underlying technology. Finally, it suggests what the elements of a corporate policy for distributed computing might be.

The Butler Cox Foundation is a research group that examines major developments in the fields of computers, telecommunications and office automation on behalf of its subscribing members. The Foundation provides a set of "eyes and ears" on the world for the systems departments of some of Europe's largest organisations.

The Foundation collects its information through its office in London and also through its associated offices in Europe and the US. It transmits its findings to its members in three main ways:

- Through regular *written reports* that give detailed findings and substantiating evidence.

- Through *management conferences* for management services directors and their senior colleagues, where the emphasis is on the policy implications of the subjects studied.

- Through *professional and technical seminars* where the members' own specialist managers and technicians meet with the Foundation research teams to review their findings in depth.

The Foundation is controlled by a Management Board whose members include representatives from the Foundation member organisations. The responsibilities of the Management Board include selecting topics for research and approving the Foundation's annual report and accounts, which show how the subscribed research funds have been employed.

Report Series No 18

DISTRIBUTED PROCESSING: MANAGEMENT ISSUES

by John Kinnear

April 1980

Butler Cox & Partners Limited Morley House 26 Holborn Viaduct London EC1A 2BP

CONTENTS

CHAPTER 1

INTRODUCTION

BACKGROUND

Early computer systems were both expensive and difficult to use. Consequently, it was not only good economics but also practical good sense to establish centralised computer systems and to support those systems by a pool of specialist staff.

Because the cost of computers has fallen and continues to fall, the economic case for centralisation is no longer clearcut. Also, the price spectrum in the computing industry is now very wide, with computers ranging in price from $100 for a basic kit to $10,000,000 for a large general-purpose commercial system. This means that an organisation can obtain equipment to fit tasks of virtually any size and at any level in the organisation. This phenomenon is really the driving force behind the move towards distributed computing, which, as we define it, is the dispersal of computing power throughout an organisation. (Indeed it could also be applied to the dispersal of computing power throughout society, but we limit ourselves in this report to the application of distributed computing within business organisations.)

Another aspect of early computer systems which encouraged centralisation was the complexity of computer operations. The consequent problems have now been largely overcome firstly, because present-day equipment is so much more reliable, and so there is now much less need for specialist staff to nurse the system constantly and to help with recovery after a system failure. Secondly, the problems arising from the complexity of operations have been greatly eased by the introduction of a wide range of software that is designed to help develop applications and to support the operation of the machine. Paradoxically, and unfortunately, this software itself has in many cases become so complex that it has generated a new need for specialist staff to install and support it, and for skilled operators to exploit its features. Nevertheless, the argument that favoured centralisation because of the specialist staff that were needed (like the argument that was based on economies of scale) is no longer as strong as it was ten or even five years ago. Nowadays, some computers can be and are operated with only occasional support from specialists.

In parallel with these changes in the nature and the capabilities of computer systems, there have been changes in social and business attitudes, and perhaps the most striking aspect of these has been the growing commitment to the concept of "small is beautiful". In line with that concept, organisations are now increasingly tending to move to decentralised management authority and, consequently, to decentralised management accountability. This trend has produced pressure from managers to be allowed to control their own computer systems. Also, in some organisations this pressure has been intensified by the managers' dissatisfaction with the level of service they receive from the centralised computer system.

This pressure from managers has produced new opportunities for suppliers who have responded by producing low-cost stand-alone computer systems, which they are selling direct to the managers concerned.

These and other developments threaten the positions and the future careers of many of the people who work in management services departments. Anybody who has spent more than just a short time in data processing is also acutely aware of the dangers of an inexpert approach to computerisation, which might lead on a wide-scale to a repetition of all the mistakes that have been made in the past. Consequently, the validity of the claims made for distributed computing is likely to be fiercely contested.

PURPOSE OF THIS REPORT

There is more than a hint of emotion in the attitudes both of those managers who are attracted to the idea of running their own computer systems, and of those management service specialists who warn of disaster if those managers are given the freedom to do so. The aim of this report is to give an objective appraisal of the benefits, the costs and the risks that are associated with distributed computing.

The concept of distributed computing is not new, and indeed by the standards of the computing industry it risks being regarded as old hat. Early distributed systems were being installed by 1970, and by the mid-70s distributed computing in its various forms was the subject of many conferences and papers. But despite all the discussion, the concept has been slow to achieve practical reality. However, most large companies have either plans or development projects in hand, but few of them have distributed systems that have been installed long enough to prove their true worth, and many organisations still rely on large centralised computers for most of their processing.

We begin in chapter 2 by providing some definitions of the concept of distributed computing. In so doing, we narrow the field of interest down to distributed processing, because we believe that this is the aspect of distributed computing that is of most interest. In the remainder of the report, then, we concentrate on that aspect.

In chapter 3 we review the arguments for and against distributed processing. Then, in chapter 4, we consider user experience to date. To a large extent, the realisation of the concept of distributed processing is being inhibited by the limitations of the currently available technology and by systems designers' lack of experience and limited ability to design and construct effective distributed processing systems. Next, therefore, in chapter 5, we summarise the technology that underlies distributed processing. Then, in chapter 6, we draw the strategic implications out of the preceding discussions and make suggestions regarding the elements of a corporate strategy for distributed processing.

READERSHIP OF THE REPORT

The report is intended primarily for those management services managers who wish either to establish or to review their strategy for distributed processing as it affects management services, or as it affects their organisation as a whole. Because the subject of the report is itself complex, the report deals with a number of complex technical issues, but it does so without going into detail. The report can therefore also be read by other senior managers in an organisation who have an interest in computing policy but have little detailed knowledge of computer technology.

CHAPTER 2

CHARACTERISTICS OF DISTRIBUTED COMPUTING

Distributed computing has been described as "putting power where the people and the business problems are". Few concerned people would disagree that it is both desirable and practicable to do this. However, there might be disagreement about just how it should be done and, in particular, about how much of the power and control should be decentralised and how much, if any, should be retained at the centre. Some distributed computing options are more radical in this respect than others, and so they present greater potential difficulties. In this chapter we discuss those options and their salient characteristics.

ORIGINS OF DISTRIBUTED COMPUTING

Most business operations are likely to involve both event-driven and schedule-driven processing. Formerly, both types of processing, if they were to be handled by a computer, would have been required to look as if they were schedule-driven. In other words, they would have been processed as batch systems. The limitations in computing technology that necessitated this have now largely been overcome and, in general, it is possible, with modern general-purpose computer systems, to cater both for schedule-driven and event-driven processing as required. Event-driven processing, which is commonly termed interactive processing, employs terminals on-line to the central system. Those terminals, depending on both their intelligence and the system requirements, may be used purely for data entry. Alternatively, they may be used to access or to update master files as well, and they may even perform simple local tasks, such as accumulating totals of cash transactions to be used for reconciliation purposes later. Sometimes, they may also be capable of continuing to operate at a degraded level when either the central system or the transmission network fails. Nonetheless, on-line terminals depend to a large extent on the central system, and effectively they operate as slaves to a central master system.

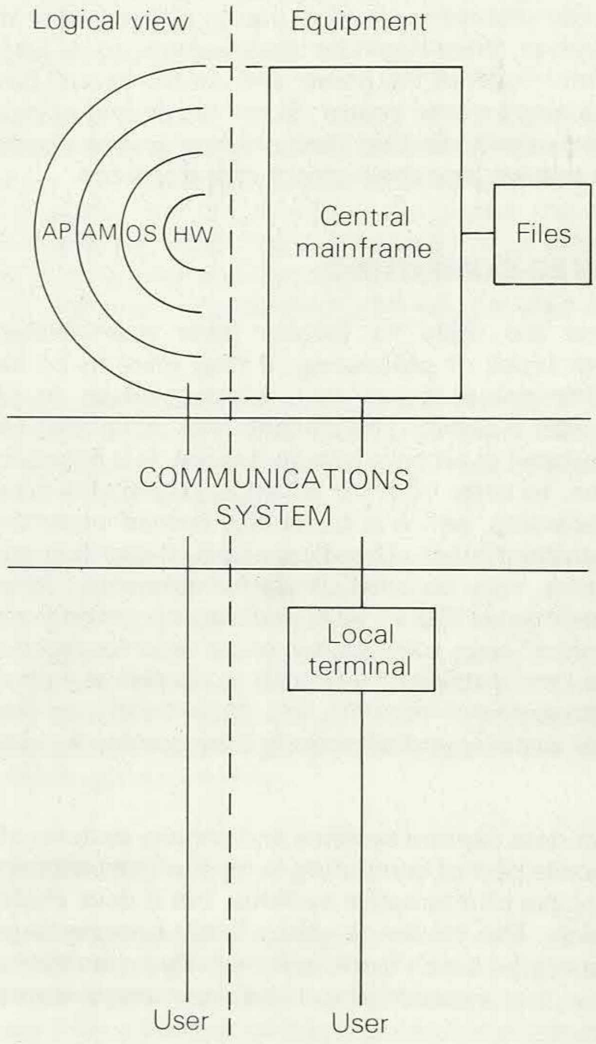
There can be no doubt that data capture systems and enquiry systems of this type will continue to be a necessary and valuable part of computing in most organisations. Distributed computing does not question the principle of interactive systems, but it does challenge the way that they are implemented in practice. The challenge arises firstly because large centralised systems, which were designed originally for batch work, and typically support a major batch load, are not ideally suited either by design or experience to handling interactive work.

Secondly, it is now often cost-effective to install a separate machine at or near the sites where the interactive terminals are installed, and that machine can be dedicated to the task of driving the terminals. This apparently straightforward extension of the principle of interactive computing is what we call distributed computing, and it conceals a number of technical, practical and organisational problems.

DECENTRALISED AND DISTRIBUTED SYSTEMS

The structure of the on-line processing systems just described — interactive terminals connected to a centralised machine — can be represented as a series of concentric rings or layers. The innermost layer is the central hardware, the next innermost is the operating system of the central machine, and the outermost is the "face" that the terminals present to the user. Each layer provides a given set of services to the next layer above it, and this cedes a degree of control in return. The whole structure, which is therefore essentially hierarchic in nature, is illustrated in figure 1.

Figure 1 On-line processing system structure



Key: AP = Applications programs
AM = Application managers for file handling, communications handling, etc.
OS = Operating system
HW = Hardware

Over the past decade, sophistications have been introduced into the structure by developments such as multi-programming (in which many applications use one operating system) and virtual machines (in which many operating systems use one machine), and by the increasing intelligence of the terminal itself. As the cost of processors has decreased, the processing power has migrated outwards through the layers. But the speed and the extent to which that migration has taken place has been constrained by many earlier design decisions. Once a certain level of control has been assigned to a particular layer the cost of re-siting it elsewhere can be very high. IBM's Systems Network Architecture (SNA), which rationalised the company's previous incoherent family of communications products, demonstrates this point. SNA cost a tremendous amount to develop, and it demands a considerable effort on the part of any user who wishes to install it. To date, it has not effected a major transfer of control away from the centre, and so the basic hierarchical structure has been maintained.

The rationale for distributed processing is that the hierarchical structure of those systems that have centralised control does not permit users enough flexibility to take their operational decisions as their business needs dictate. To provide users with sufficient flexibility appears to demand a local system that is capable of autonomous operation and that has its own operating system. Sometimes, this requirement for local flexibility can be strong enough to break completely the umbilical cord that links the site or the department to the central system. As a consequence, links with the centre become casual in nature, and the local site uses any convenient means to send summary data to the centre, and the centre similarly uses any convenient means to send instructions to the local site. The user sees the local system and the centralised system as being two entirely separate systems. We term this arrangement decentralised processing. Figure 2 illustrates the structure of decentralised processing systems and shows how it is different from the structure of on-line processing systems.

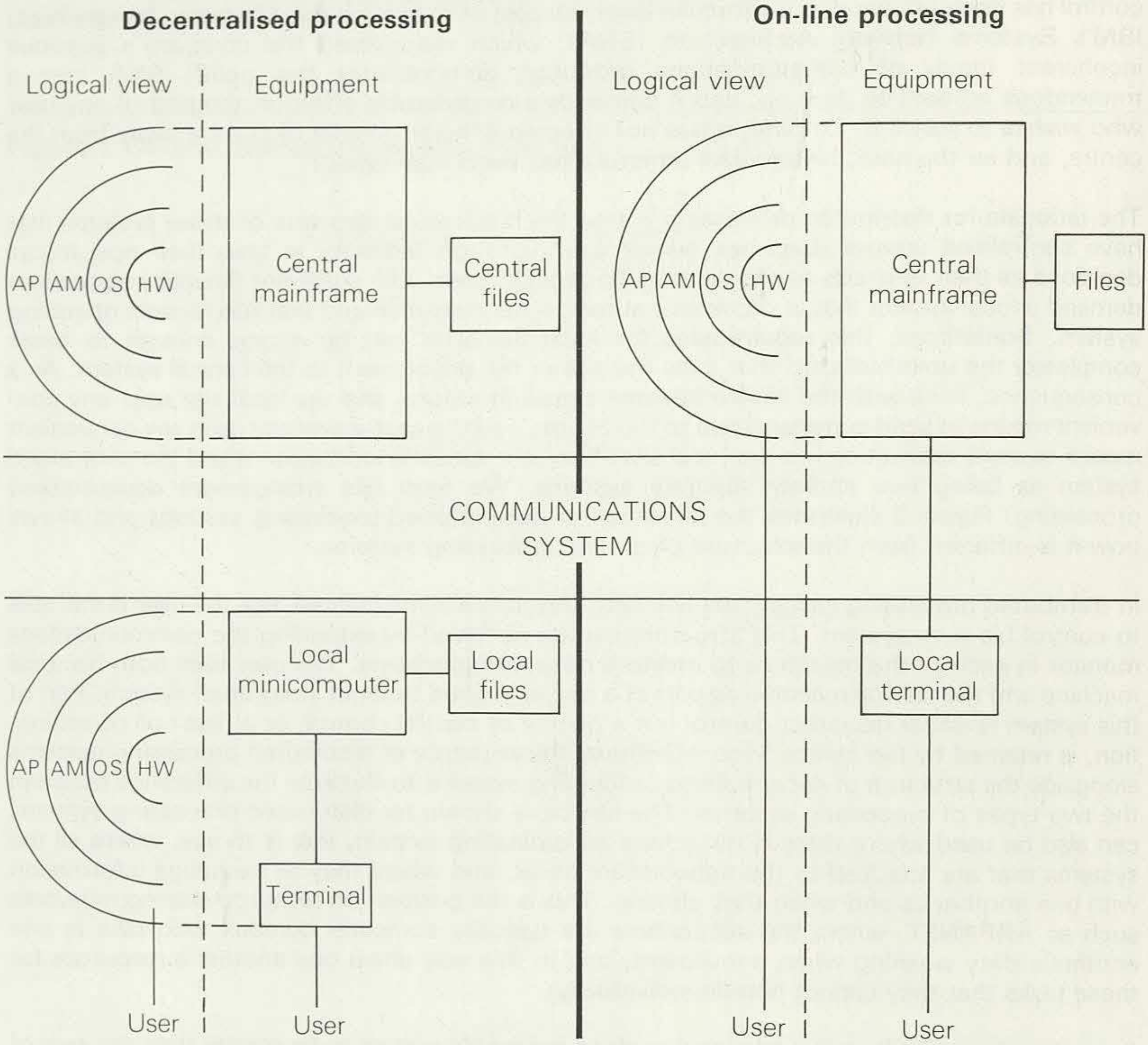
In distributed processing proper, the link with the centre is maintained, but the user is still able to control his own system. This structure can be achieved by extending the communications monitor in each of the machines to create a networking scheme. The user sees both his local machine and the central machine as part of a single, unified (but not integrated) system. Part of this system is under his direct control but a degree of central control, or at least of co-ordination, is retained by the centre. Figure 3 shows the structure of distributed processing systems alongside the structure of decentralised processing systems to illustrate the difference between the two types of processing systems. The structure shown for distributed processing systems can also be used where there is no central co-ordinating system, that is to say, where all the systems that are attached to the network are equal, and where they all exchange information with one another as and when they choose. This is the position on resource-sharing networks such as ARPANET, where the subscribers are typically computer bureaux who take in one another's dirty washing when convenient, and in this way share one another's resources for those tasks that they cannot handle individually.

In a sense then, distributed processing enables the whole system to be greater than the sum of its parts. The risk with decentralised processing is that the whole system becomes less than the sum of its parts, because there is no co-ordination. On the other hand, this arrangement does have the great merit of simplicity.

IMPLICATIONS OF DISTRIBUTED PROCESSING

Although we have drawn a distinction between decentralised processing and distributed processing, the former is really only a special case of the latter. If all the systems in a decentralised scheme are genuinely independent of the centre, and if they supply no corporate or summary information and require no services or instructions, then they are effectively stand-alone systems and can be treated as such. They may present problems because they are developed and operated away from the centre, but similar problems arise also with distributed processing.

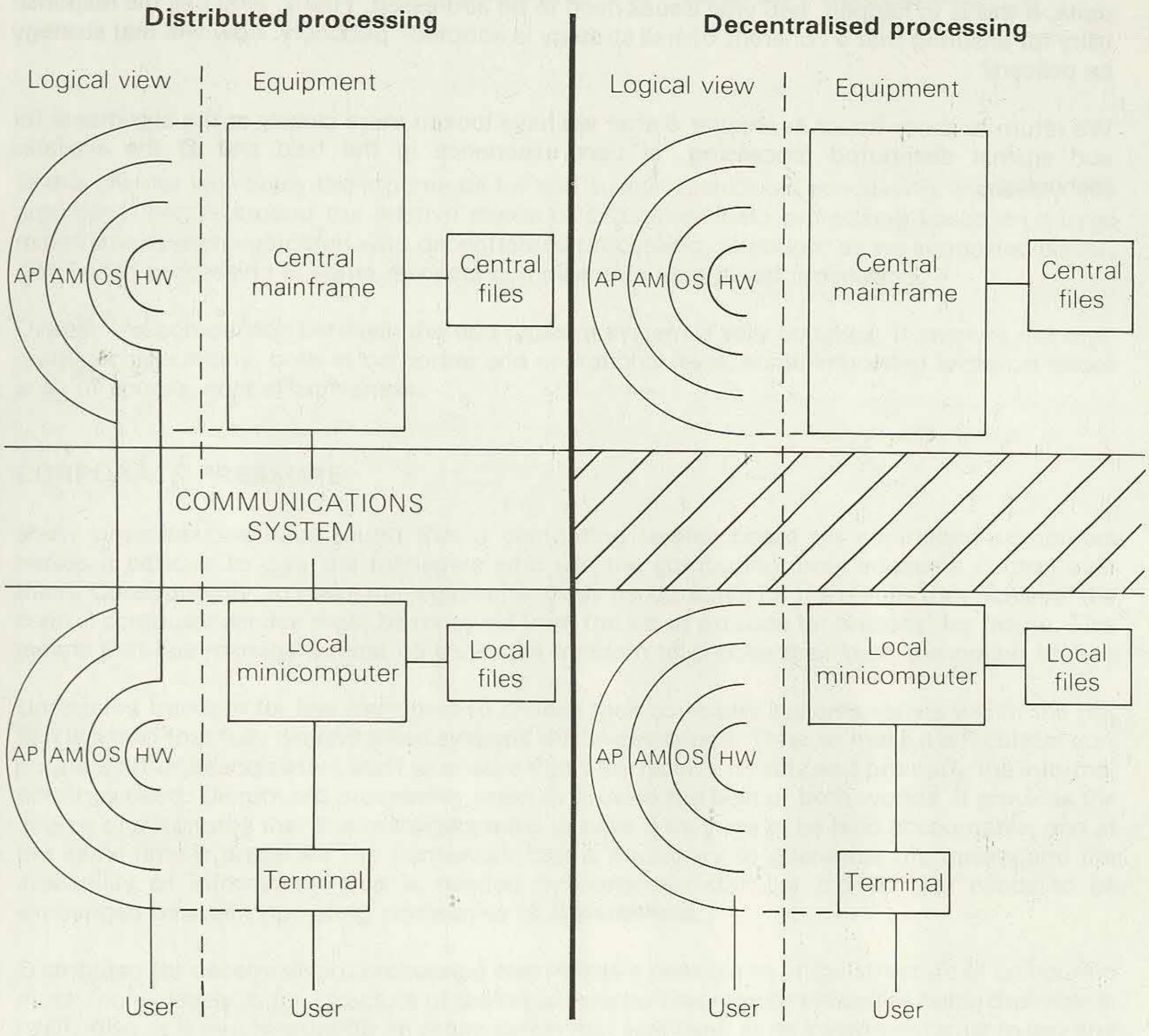
Figure 2 Decentralised processing and on-line processing system structures



Key: AP = Applications programs
AM = Application managers for file handling, communications handling, etc.
OS = Operating system
HW = Hardware

Equally, an organisation may adopt a decentralised processing solution as an expedient because it judges that the technology and the available skills are not capable of supporting the more sophisticated distributed processing option. Such an approach is perfectly valid, but if an organisation views decentralised processing as a stepping stone to distributed processing, then it presumably needs to have a policy for distributed processing. Clearly, the evolution from decentralised processing to distributed processing is less likely to take place by accident than by plan.

Figure 3 Distributed processing and decentralised processing system structures



Key: AP = Applications programs
AM = Application managers for file handling, communications handling, etc.
OS = Operating system
HW = Hardware

Because of what has been said above, we concentrate our attention in the remainder of this report on distributed processing, as we have defined it earlier in this chapter, and we assume that it includes decentralised processing. We have defined distributed processing mainly in technical terms, but its implications extend well beyond the mere technical realm. Distributed processing raises a number of questions, and the answers to those questions will have an enormous impact on the way in which an organisation develops its future use of computing technology.

Distributed processing obviously means that some of the responsibility for operating computer systems will be decentralised. By extension, this may also mean that the responsibility for developing and maintaining computer systems will follow the hardware out to the operating units. If this is to happen, two vital issues need to be addressed. Firstly, who has the responsibility for ensuring that a coherent overall strategy is adopted? Secondly, how will that strategy be policed?

We return to these issues in chapter 6 after we have looked more closely at the arguments for and against distributed processing, at user experience in the field and at the available technology.

CHAPTER 3

ARGUMENTS FOR AND AGAINST DISTRIBUTED PROCESSING

In this chapter we review the arguments for and against distributed processing. Inevitably, the arguments centre around the relative merits of centralised data processing based on a large mainframe system compared with decentralised processing, although, as we suggested earlier, distributed processing in some respects complements centralised processing.

Overall, the comparison between the two types of system is very complex. It involves management considerations, both at corporate and operational level, some important technical issues and, of course, cost-effectiveness.

CORPORATE PRESSURE

Many organisations have found that a computing service based on centralised computers makes it difficult to give the managers who use the computing tools adequate control over them. Consequently, to make managers effectively accountable for the results they achieve, the central computer service must be removed from the list of excuses (or reasons) for failure. This means that line managers must be given the freedom to choose their own computing tools.

Unfettered freedom for line managers to choose their computer systems carries with it the risk (if it is a risk) that fully decentralised systems will be developed. This can make it difficult for corporate staff or headquarters staff to ensure that they receive reliably and promptly the information they need. Distributed processing seeks to provide the best of both worlds. It provides the degree of autonomy that line managers need to have if they are to be held accountable, and at the same time it preserves the framework that is necessary to guarantee the quality and the availability of information that is needed by corporate staff, or indeed that needs to be exchanged between operating companies or departments.

Distributed (or decentralised) processing also makes it possible to fit the structure of computing much more closely to the structure of the organisation. This may be viewed as being desirable in itself. Also, it is much easier for an organisation that sees itself as an investment trust to buy and dispose of assets if those assets come complete with in-house computer systems (although it is always possible to continue to service requirements for centralised data processing on a bureau basis).

However, although distributed processing ostensibly appears to offer organisations greater flexibility in the way they configure and modify their structure, some caveats are necessary. It has been argued that it is easier to modify a central database (a logical model of the enterprise) than it is to modify the physical structure of computer systems, whether they are distributed or otherwise. However, this is only a theoretical possibility at present, because very few, if any, organisations have a complete enough model of their enterprise on their database for it to be used in this way.

USER PRESSURE

End-users of computer services have probably wanted to have their own computers since the earliest days of computing, and so there is nothing new about users' desires for ownership. What is new is that line managers are now more likely to persuade themselves that it is worth the risk, even though most of them are sophisticated enough to understand the implications of ownership, and probably have no real desire to subject themselves to the headaches that are involved in managing one's own shop.

But once line managers come to regard the central service as a near monopoly with a captive market, and with no strong motivation to provide its customers with good service, their desire to improve matters may well override their reservations. If this poor service motivation is supported by a strong cost motivation — as it easily can be — it can quickly become irresistible.

Some users whose data is held on centralised databases are also concerned about the lack of control they have over that data. Centralised large-scale systems are likely to incorporate more sophisticated mechanisms for archiving data, for preventing unauthorised intrusion and for recovering data after a system failure, than a small local system, for example, would. This does not necessarily guarantee that data will be either more secure or better protected against intrusion, because both these aspects depend on the procedures that are adopted to achieve them. The Ford Motor Company, for example, found that data integrity improved on small local systems. Data integrity also depends to a large extent on the potential source of violation. There will certainly be cases where private safes at the operating unit are more secure than is a closely guarded central site, and vice versa.

Users' desires to control their own systems or to control their own data are rooted in the psychological realities of people's attitudes to the tools they use. The desires are not necessarily or solely linked to the technical merits of decentralised processing versus centralised processing. However, there are some respects in which small, local systems offer definite operational advantages over the centralised approach. In particular, local systems can offer users:

— Interactive data capture with full integrity and validity checking.

— Easy access to files maintained locally, for handling enquiries and preparing reports.

— Easy rescheduling to meet special needs or emergencies.

— More consistent response times and levels of service.

The above benefits are not confined to the use of small, local systems, but there is evidence that they are more easily obtained by using those systems than by using centralised systems with interactive terminals. On the other hand, small, local systems certainly do not perform some tasks as efficiently as large-scale centralised systems do, because their operating software is less mature. In particular, these are the mundane housekeeping tasks, such as archiving, security logging, etc. Those tasks tend not to be visible to the user except when things go wrong, and so they are not always given the attention they deserve. However, since a small, local system is often a dedicated resource, much less elaborate procedures for security and fall-back can be used with it. For example, provided that failures are infrequent, it will often be possible to recover from a failure of a small system by re-entering all the current day's transactions. It would be unthinkable to use such an approach with most centralised systems.

SPEED OF IMPLEMENTATION

It seems that the dedication of the resource is also one of the reasons why applications on small, local systems have a reputation for being simpler (and so, quicker) to develop and install. A strongly motivated line manager does not need to enter into tedious negotiations to get programmers assigned to him or to get local management support for what he wants to do, because he can merely go out and obtain for himself the resources he needs.

The following two technical factors also tend to reduce the initial effort that is necessary to implement a small, local system using a minicomputer:

- The operating systems are designed to achieve only limited objectives, and so they are cleaner and easier to work with.
- It takes comparatively little administrative effort to fit the new application into the existing systems environment. For example, retuning the operating system, extending or modifying the files, and allocating disc space, etc., are all straightforward operations.

The net effect with a small, local system is that applications can be implemented more quickly. This is a great advantage, even if it may mean that some problems are postponed rather than avoided altogether.

However, the most significant reason why small systems can be implemented quickly appears to be a matter of attitudes rather than a matter of technology. Quite simply, when users are in control of their own affairs they set lower levels of ambition. This can result in poor documentation or poor standards in general (which are by no means perfect in large centralised installations either), but more importantly it means that the priority in designing systems is to meet the essential needs first and other needs later. As Pareto's Law states, 80% of the needs can often be met for 20% of the effort. When the user largely has to do the work himself, he will inevitably concentrate on what appears to him to be the essential aspects.

By contrast, a user when faced with a systems analyst who is attempting to determine the requirements for an application to be implemented on a centralised system, is under pressure to state all his requirements and to get them right. He knows that subsequent modifications will be expensive to make and will probably take an inordinate amount of time to implement. He will naturally assume that a large centralised machine can do whatever he wants, whereas he knows that the small system has a natural, physical boundary beyond which he cannot afford to stray. As a consequence, centralised systems can easily be excessively elaborate from the outset. This means that they take longer to implement, and this increases the probability that his requirements will have changed by the time the system is up.

Modern software tools, such as database management systems and high-level languages like APL, are beginning to address this problem by enabling systems to be built more quickly, but it is doubtful whether they will outweigh entirely the advantage of the small system.

To some extent, the real advantage in the speed of implementation lies with stand-alone decentralised systems. If an organisation wishes to implement distributed processing quickly it needs to have an existing communications infrastructure that will permit linkage with the centre or with other related systems. Failing that, it will at least need to have a set of protocols and rules governing communication that it can easily adopt. Neither of these will necessarily exist at the outset.

EQUIPMENT COSTS

The cost-effectiveness of distributed processing solutions depends heavily on the ratio of the cost of data transmission and the cost of computer processors. Distributed processing places more processing power at the local site. Consequently, it reduces the need to communicate, because it is necessary to transmit only summary information and not the individual raw transactions. Because the cost of processing is falling more quickly than the cost of data communication is, time favours distributed processing. This argument has equal force for on-line systems with intelligent terminals that can compress the data. Distributed processing is, however, a more radical solution, and so it offers more scope for achieving benefits.

What has been said above may represent the theoretical case for distributed processing, but the position in practice is not so straightforward. Some case histories show that distributed processing can cause equipment costs to rise as well as to fall. Distributed processors clearly can reduce communication with the centre to the minimum that is necessary. They can also make more intelligent use of the facilities that are available, for example, by batching data for transmission at cheap overnight rates. Yet, in some circumstances, distributed processing can lead to increased costs. If, for example, it is necessary to replicate part of the central files at each local site, this requirement may well represent a greater communications cost than would be incurred with terminals on-line to a single central database. Distributed processing could also call for new lateral connections between the several sites, in addition to the connections that link each site to the centre. It is, therefore, more realistic to view distributed processing as a means of increasing system effectiveness, rather than as a means of reducing the overall equipment costs.

Provided that the measure used is not raw power alone, but the amount of useful work that a system can perform, there is mounting evidence to suggest that economies of scale no longer apply to computer equipment costs. This change has come about firstly, because the software overhead in a large system that handles on-line terminals can be very high (as is discussed further on pages 13 and 14). It has come about secondly, because the processor now accounts for a smaller and smaller proportion of the total cost of a typical configuration.

In a distributed processing system, mainframe processing and storage capacity are replaced by a number of smaller processors, each of which has local storage. Because, as is typically the case, the mainframe processor and storage represent only 30% of the total system cost, there is limited scope for either saving or spending more in this area alone. The overall cost depends much more on peripherals and terminals (which represent 70% of the total cost) where there will be only limited replication of equipment.

One other cost area in which the central mainframe appears to be inferior to distributed processors is in its ability to handle incremental growth. Whereas individual small systems can be upgraded progressively at relatively low cost, to upgrade a mainframe involves large cost step functions. It is easy, however, to overlook the potential consequences of growth within a distributed processing system. Growth can sometimes place unanticipated demands on the communications network, and sometimes it can lead to the very costly restructuring of programs and/or data if the upper limits of a small system's capacity are overstepped. The lesson appears to be that distributed processing systems should not be implemented either on a hardware configuration that is close to its performance limits, or in a way that places high demands on the communications system. High growth applications will also be risky.

PEOPLE COSTS

Staffing costs are incurred in two main areas:

- Operation and support.
- System development and maintenance.

Distributed processing can affect costs in both these two areas. We discuss below what the impact might be.

1. Operation and support
Individually, small systems demand little effort to operate because they are run at much

lower levels of utilisation than are large-scale systems. Consequently, there is no requirement for highly-skilled specialists to operate or tune the system. (Often, 50% of staff costs at centralised systems are in operations, although this proportion is decreasing as data entry becomes interactive and as equipment becomes more reliable.) Sometimes, the operation of a small system can become part of the duties of those staff who use it, and so it becomes more of a marginal cost than a straight overhead. This tendency also makes comparison with a central mainframe difficult, because these and other tasks that are carried out by full-time operators at a central site are dispersed (and so are concealed) at the user site. Often though, this breakdown of traditional demarcation lines, as with some other industries, can lead to higher productivity overall.

If there are potential reductions in operating costs to be had from distributed processing, there are also risks. Once a distributed system becomes so complex as to require dedicated operators and dedicated systems programmers at every site, the cost can become very high indeed.

2. System development and maintenance
   Most organisations have a substantial investment in mainframe-based applications and also in the skills that go with them. This investment is itself a disincentive to implement distributed processing, which will often demand new skills. This disincentive may be reason enough for management services to settle for the benefits that can be derived from using on-line terminals without distributed processing, particularly since distributed processing may involve unfamiliar equipment and unfamiliar technical problems.

   At the moment, the mainframe approach also has another advantage. A much wider range of packaged applications software and software aids is available for mainframes than is available for small systems, and particularly for minicomputers. However, this advantage is gradually disappearing. But despite this present advantage, a user may well be more ready to accept a package on a minicomputer that he can buy for himself, than he will be to accept a mainframe package that is "foisted" on him by management services. Emotional judgement often has a greater influence on decision making than does rational judgement.

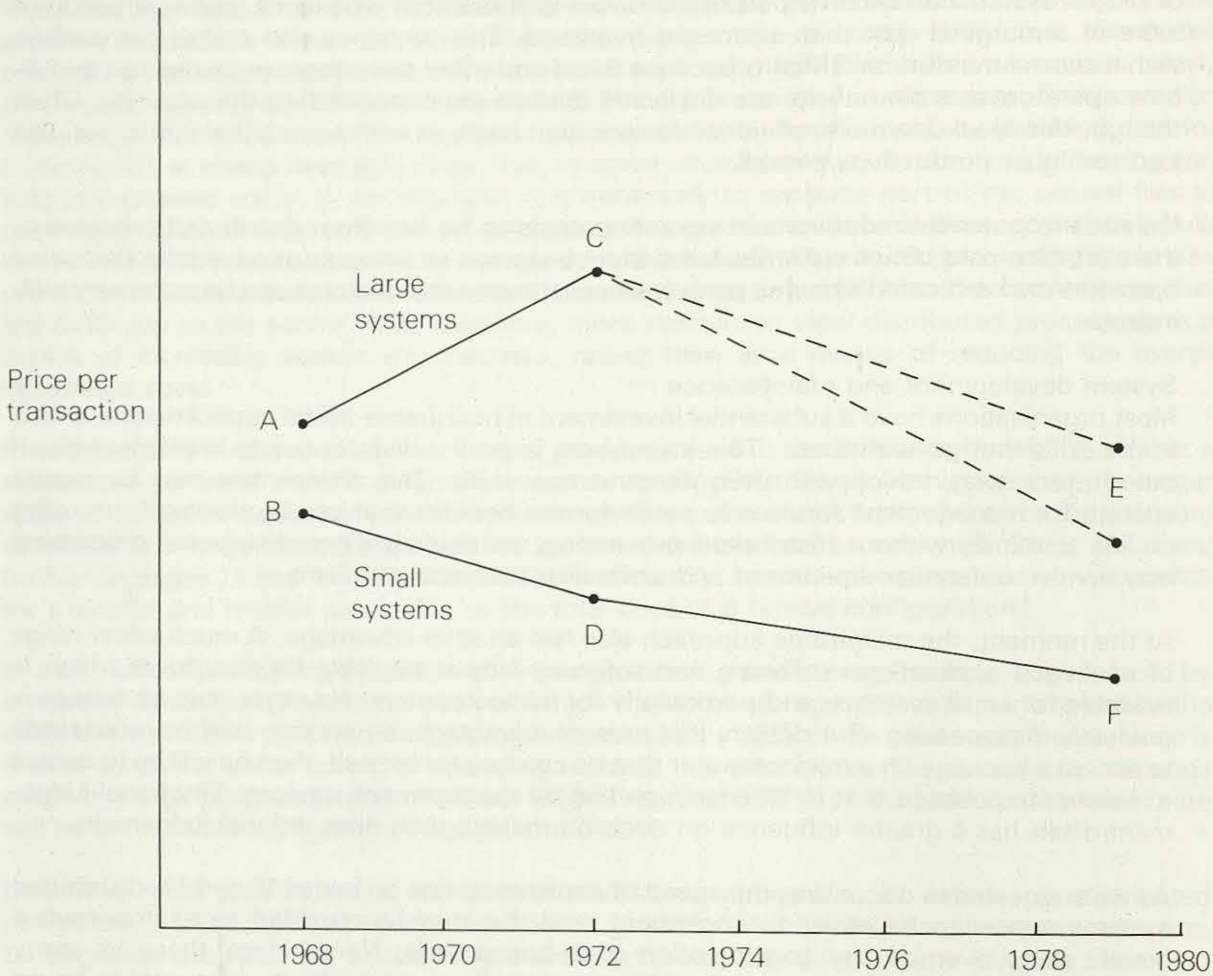
   As we suggested in discussing the speed of implementation on pages 10 and 11, distributed systems appear to be easier to implement, and this may be regarded as so important a benefit as to override any consideration of in-house skills. Nonetheless, there are many dangers in developing distributed processing applications without adequate in-house control. It can lead to duplication of effort, to dependence on outside contractors, to severe problems of maintenance and support and to a lack of continuity in corporate systems. Clearly, these represent key issues that a corporate strategy needs to address.

OTHER ISSUES

One or two technical issues have been touched on earlier in this chapter. Below we discuss briefly the claims and the counterclaims that are made for and against distributed processing on a number of technical counts. We also discuss the vulnerability of centralised and distributed processing systems.

1. Performance
   Figure 4 shows a performance comparison between a large-scale system that drives on-line terminals, and a small, local system. This comparison originates from the IBM Systems Research Institute in New York, and it shows the evolution in performance from 1968 to the present, in large-scale systems (line A-C-E) and small systems (line B-D-F). The author attributes the "astonishing" upward slope of the large system line from 1968 to 1972 mainly to systems software overhead (H. Lorin, "Distributed processing: An assessment", *IBM*

Figure 4 Performance comparison of large and small systems



(Source: H. Lorin, IBM Systems Research Institute)

Systems Journal, Vol. 18, No. 4, 1979). This has aptly been described as the Pyrrhic victory of software over hardware that is designed to handle serial, batch work rather than parallel, transaction-oriented tasks. The figure shows that, to the present day, small systems continue to enjoy an advantage over large systems in driving terminals. This finding reflects general user experience also.

It is not very surprising that this should be so, bearing in mind that the software path length of a transaction on a minicomputer is typically less than 1,000 instructions, whereas on a large computer with a sophisticated operating system it is often greater than 100,000 instructions.

2. Stability

System software on minicomputers is normally more stable than is system software on general-purpose mainframe systems. Mainframe suppliers have traditionally adopted an evolutionary approach to software development, and have carried their users with them from one hardware generation to the next, and from one feature to the next. This evolutionary approach produces a continuing flow of modifications to the software. Unfortunately, those users who do not need the new features are compelled to introduce the modification, because they fear that they may fall so far behind as to be unable to catch up at all. Now, however, the mainframe suppliers are beginning to offer additional optional features on top of a relatively stable basic operating system. Undoubtedly, though, the fundamental problem will persist for some time.

Minicomputer suppliers, on the other hand, have a less diverse market to serve. Consequently, they tend to issue more radical software updates but they issue them far less frequently than do the mainframe suppliers. This difference of policy raises the question of whether it is better to update the software continually or to drastically re-vamp it occasionally. The current pace of technological change and the relatively short useful life of some applications favour the latter approach. Equally, database applications, which are designed to survive detailed changes in requirements, might be better suited to the former approach.

3. Reliability
The way in which distributed systems are designed tends to make them inherently more reliable because:

- A failure at one distributed site does not affect operations at any of the other sites.
- The operating software and the applications software are dedicated to the task in hand.
- Applications systems tend to be simpler and to have cleaner interfaces with one another.

Hardware failure rates are comparatively high on minicomputers at present, but minicomputers will undoubtedly become more reliable as a result of higher levels of circuit integration. However, the impact of a failure tends to be less with a minicomputer because it is localised. By contrast, a mainframe failure can cause all departments to lose service, and it is then far more difficult to borrow staff to help with recovery, since all departments have the same simultaneous problem.

Some failures will not be solved with unskilled support alone, and in this respect the local minicomputer is more vulnerable. Often there will be no expert on hand to diagnose the problem, and perhaps to repair the trouble, as there probably will be at the central site. There may also be a longer delay in providing engineering support at a local site, particularly since some central sites have a resident engineer. On the other hand, it must be borne in mind that staff who are thrown back on their own resources when things go wrong will more quickly become adept at recognising the nature of problems and at taking appropriate recovery action than will staff who are faced with an opaque, remote system.

Suppliers are also building facilities into their distributed systems to enable errors to be diagnosed remotely, so that some maintenance and support can be supplied on a centralised basis.

4. Fail-soft
The issue of reliability is closely tied up with the impact of failures. It is particularly tied up with the ability of the system to fail softly (that is, to fall back to a degraded level of service or to call in alternative processing facilities on failure). Distributed processing systems will normally be, and should be, designed to continue to supply essential services if a failure occurs in either the communications system or the central mainframe. Also, when each

Figure 5 Summary of the merits and demerits of the alternative system structures

<table>
<tr><th>Decentralised processing</th><th>Distributed processing</th><th>Centralised processing</th></tr>
<tr><td>Threatens corporate and internal communication</td><td>Demands an initial technical effort</td><td></td></tr>
<tr><td colspan="2">Promotes management accountability</td><td></td></tr>
<tr><td colspan="2">Fits computing to the organisational structure</td><td></td></tr>
<tr><td colspan="2">Gives the user control of his own systems and data and better on-line performance, but involves him in new tasks</td><td>Permits a more professional approach</td></tr>
<tr><td colspan="2">Shortens system development cycle</td><td>Provides more advanced tools and techniques</td></tr>
<tr><td colspan="2">Provides economies of dedication</td><td>Provides economies of scale</td></tr>
<tr><td colspan="2">Can be upgraded progressively</td><td>Has large cost step functions</td></tr>
<tr><td colspan="2">May not handle rapid or unexpected growth</td><td></td></tr>
<tr><td colspan="2">Is easier to operate</td><td>Requires dedicated operations team</td></tr>
<tr><td colspan="2">May become more expensive to operate if complexity grows</td><td></td></tr>
<tr><td colspan="2">Demands new design and programming skills</td><td>Offers a wide range of packaged software</td></tr>
<tr><td colspan="2">Has more stable but less mature software</td><td>Demands frequent software modification</td></tr>
<tr><td colspan="2">Is likely to be more reliable and disperses the impact of failures</td><td>Is vulnerable</td></tr>
<tr><td></td><td colspan="2">Encourages better control of development</td></tr>
</table>

Note: Demerits are shown in italics.

component of the system handles a proportion of the overall load, it is easier to replicate vital pieces of equipment. Replication can also be applied more selectively where it is really needed. But, for a very high availability system, where each component needs to be replicated, a single highly reliable system may prove to be the better choice.

5. Vulnerability

   Any failure of a centralised processing system will potentially affect all the users of the system. A centralised system is therefore vulnerable to any event which may cause loss of service, including:

   - Systems failures.
   - Hardware failures.
   - Sabotage.
   - Industrial action.

   Distributed processing systems minimise the risks associated with those events, since the occurrence of a failure is unlikely to affect simultaneously all of the distributed sites.

SUMMARY

Historically, there have been overwhelming technological and other reasons for favouring centralised batch processing. Now, however, many of these reasons have ceased to apply or else have lost much of their force. The technology now gives both the system designer and the user a choice between decentralised, distributed, and centralised solutions. We summarise, in figure 5, the main merits and demerits of the three different approaches, based on the discussion in this chapter.

Distributed processing stands between centralisation and decentralisation. The argument for distributed processing is not that centralisation is better than decentralisation, or vice versa. It is rather that decentralisation needs to be managed at a technical level, if it is to deliver long-term as well as short-term benefits in computer systems. Because it places control of computer systems directly in the hands of users, distributed processing also needs to be managed both at a human and an organisational level. We discuss the detailed implications later in this report.

CHAPTER 4

USER EXPERIENCE

Many successful examples of distributed processing systems now exist. Most of these are interesting from a technical point of view, and they demonstrate conclusively that distributed processing works, and also that it can be highly cost-effective. Beyond that, however, it is difficult to draw any general lessons on either the right or the wrong way to approach distributed processing. In this chapter we describe two approaches to distributed processing that do offer clear lessons, and we also summarise experience in general.

TWO CASE HISTORIES

The two case histories that we describe below are interesting for two reasons. Firstly, they both represent a committed root-and-branch approach to the decentralisation of computer power. Secondly, they both provide experience of the practical implications of distributed processing that can be said to have stood the test of time.

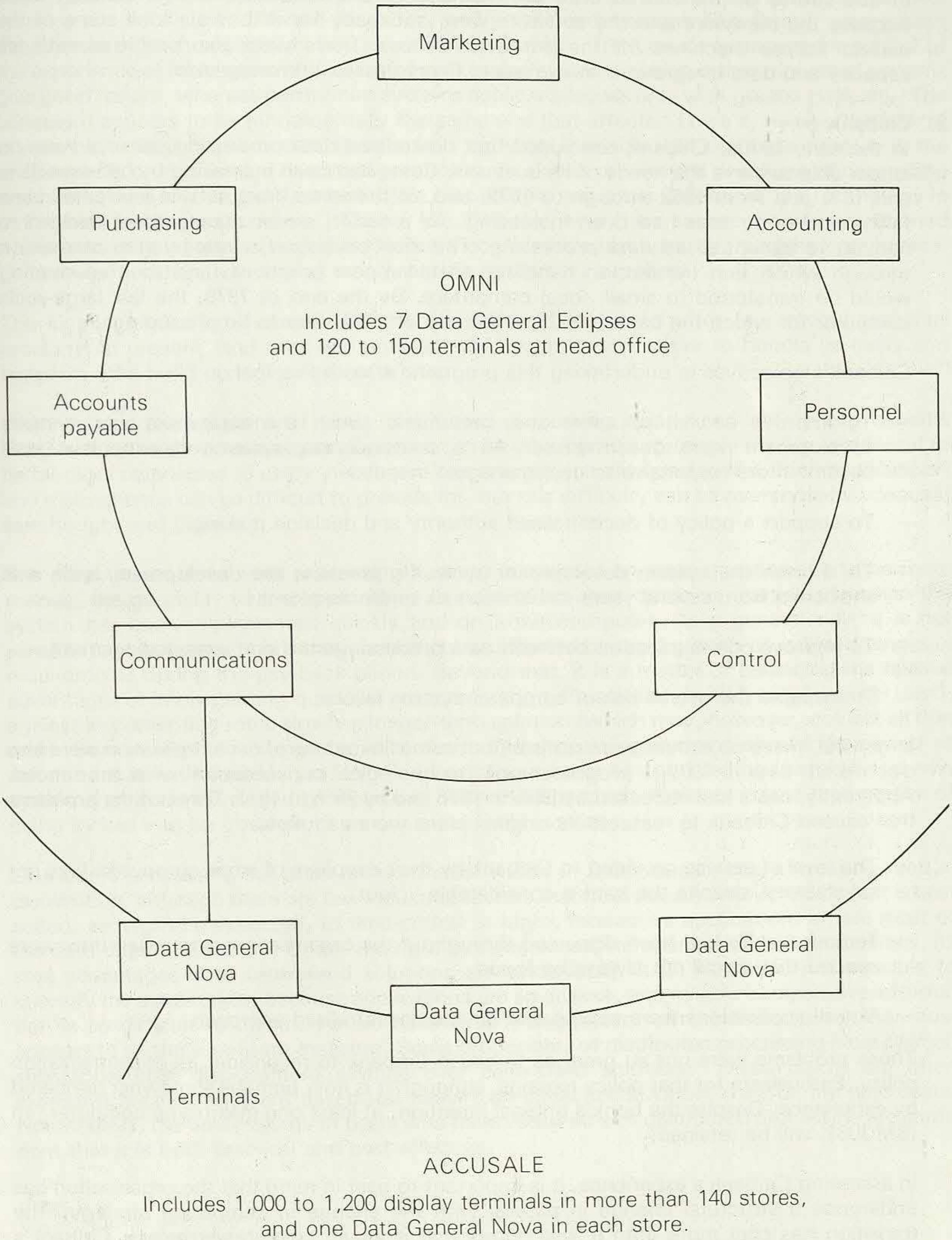
1. Lowe's Companies Inc.
   The first case history concerns a US retailer, called Lowe's Companies Inc. In 1973, Lowe's implemented distributed processing at store level, with a system called Accusale. Briefly, Accusale is a terminal-based sales and inventory control system, that operates on a stand-alone basis in the store during the day, and transmits sales and inventory information, unattended, to the corporate office overnight.

   By 1975, hardware had been installed in over 40 of the 140 stores, and each installation consisted of a Data General Nova 1200 minicomputer with a 4 Mbyte exchangeable disc and up to 16 visual display terminals for sales staff to use. No dedicated operator was required — indeed, the system was claimed to require no intervention on the part of people in the store. When technical support was required, it was brought in from the centre. Up-time, after the first six months of operation, was found to be 99.45%. Development had taken one man-year and an elapsed time of six months. Benefits, according to store managers, were a 30% improvement in the productivity of sales staff, and improved control information.

   The second part of Lowe's distributed processing plan was a corporate system to replace the existing IBM mainframes. The new system, called Omni, was to consist of a number of Data General Eclipse C/300 minicomputers linked in a ring network. Each department would have its own dedicated machine, and there would also be a communications machine to link in the Accusale in-store systems, and a control machine to provide co-ordination and monitoring of the whole configuration (which is shown in figure 6). Lowe's expected that Omni would give a 20% decrease over current data processing costs, plus the benefits that individual departments would derive by having on-line access to what was effectively a dedicated machine.

   The position in late 1979 was as follows. The Accusale system had been installed in over 200 stores and was viewed as a great success. Field maintenance is run from a number of bases and up-time has been maintained at a level that exceeds 99%. Minor software changes are down-line loaded by a central machine to each site, using dial-up telephone lines. Any major changes that involve a degree of retraining are installed on-site by the maintenance team.

Figure 6 Lowe's planned distributed processing systems



Omni, on the other hand, has not been a success. There were a few stand-alone systems still running on the Eclipse C/300 machines (payroll, accounts payable) that provide data for the central accounting package on the mainframe (an IBM 3031) by tape transfer. These will in due course be transferred onto the mainframe. It is believed that the concept failed because the hardware and the software were not ready for it (Lowe's took some of the earliest Eclipse machines off the production lines). There were also problems with file capacity and data integration. In summary, Omni proved unmanageable.

2. Citibank

In the early 1970s, Citibank concluded that centralised data processing services were no longer able to serve the needs of its business. Costs had been increasing by between 15% and 18% p.a. from 1962 through to 1970, and, at the same time, service levels had been falling and error rates had been increasing. As a result, senior management decided to convert to decentralised data processing. The plan envisaged a step-by-step conversion through which first transaction handling, then support functions, and then accounting would be transferred to small, local computers. By the end of 1979, the last large-scale computer (of which the bank had 25 at the end of 1974) was to be phased out.

Citibank's objectives in undertaking this programme were five-fold:

— To improve service to users and customers. (Both the equipment and systems development were decentralised. As a consequence, systems development staff became more responsive to user managers' needs.)

— To support a policy of decentralised authority and decision making.

— To shorten the system development cycle. (In practice, the development cycle was shortened from several years to between six and nine months.)

— To improve communication between data processing staff and user management.

— To disperse the overall risk of computer system failure.

In general, these objectives were realised, but some limitations of decentralisation were also discovered. Late in 1977, a programme of "technological consolidation" was announced. Apparently, costs had increased by 23% in 1976 and by 26% in 1977. Three of the problems that caused Citibank to reassess its original plans were as follows:

— The level of service provided to Citibank by their suppliers of small computers was not satisfactory, despite the bank's considerable "clout".

— Technical skills had been dispersed throughout the organisation, and when they were needed they could not always be found.

— Not all applications were appropriate for the decentralised approach.

Those problems were not so great as to cause Citibank to reconsider its decentralisation policy. Enthusiasm for that policy remains, although it is now perhaps somewhat tempered by experience. Despite the bank's original intention, at least one mainframe computer, an IBM 3031, will be retained.

In assessing Citibank's experience, it is important to bear in mind that the organisation has undergone a structural change in parallel with the change in computing strategy. The transition has cost more than it was hoped that it would cost. Nevertheless, Citibank's achievement is a striking vindication of distributed processing, even though it provides some warnings.

GENERAL EXPERIENCE

The Lowe's case history illustrates the two different types of distributed processing systems. These respectively are replicated systems (like Accusale), where a number of similar or identical systems are installed in different locations, and partitioned systems (like Omni), where each department or each application has a dedicated machine. The Lowe's experience is typical of the experience of most users of distributed processing systems, which is that replicated systems give good results, whereas partitioned systems achieve success only with greater difficulty. The constraint appears to be fundamentally the same one that affected Lowe's, namely the lack of mature technology. In particular, there is little software available that can cope either with the problems of communication in a configuration such as this or with the problems of controlling and managing a distributed database. (We discuss this and other aspects of the technology in more detail in the next chapter.) Those, like Citibank, who have succeeded with partitioned systems have generally been able to devote a considerable in-house effort to the communications system, rather than relying mainly on the equipment suppliers.

The simpler replicated systems, on the other hand, appear to offer few severe technical problems at present, and they will undoubtedly become even easier to handle as users and suppliers alike build up experience with them.

From an operational point of view, distributed processing clearly can deliver the benefits discussed in the previous chapter. Systems can be developed surprisingly quickly, are found to be far more responsive to users' needs and can operate virtually unattended. Technical support and maintenance can be difficult to provide for, but this difficulty can be overcome by adequate forethought and planning.

The durability of distributed processing systems is more of an open question. Or to be more precise, the durability of the applications is more of an open question, particularly when the system has been implemented quickly and on a minicomputer. As a general rule, it is not sensible to select applications that are likely to be affected by rapid growth or radical changes in requirements during the pay-back period. Beyond that, it is a matter of balancing the relative advantages of implementing quickly a potentially short-life solution (which may however, last), against implementing more slowly a longer-term solution (which may, however, not last all that well). Some users have suggested that distributed processing systems should be viewed as solutions that have limited lives. When an existing system has outlived its usefulness, a new system should be developed to replace it. With this approach, the user would face no risk of being locked into an obsolete or an unmaintainable system.

Distributed processing has been adopted most widely in financial services, in retailing, and in distribution, although there are few industrial sectors that now remain untouched by it. It is best suited, as might be expected, to time-critical or highly interactive applications where most of the data originates locally. Under the right circumstances, its operational and, sometimes, its cost advantages over centralised solutions are undeniable. It is, however, not possible to quantify the advantages, because applications are so diverse, and reliable comparative information is so difficult to obtain. As we have already suggested, effectiveness rather than cost appears to be the significant measure. Some proponents of distributed processing have claimed substantial cost displacement savings over centralised solutions. These claims are often excessive because they are based too heavily on initial costs, rather than on life-time costs. Nonetheless, the vast majority of users who have ventured into distributed processing are confident that it is both practical and cost-effective.

CHAPTER 5

THE TECHNOLOGY OF DISTRIBUTED PROCESSING

Although many distributed processing systems have already been constructed, and many others are now being constructed, the technology is not yet mature in all respects. In this chapter we discuss the major elements of a distributed processing system and the capabilities of the available products. We do not, however, attempt to provide a comprehensive product review, because the market is a broad one and is evolving quite rapidly. We have chosen examples that illustrate the points that we make in this chapter, but they are not necessarily the best or the only ones in their respective fields.

We now discuss the present state of the art as far as distributed processing is concerned, in processors and operating software, in file management and in communications. We then discuss the impact of microelectronics in relation to distributed processing.

PROCESSORS AND OPERATING SOFTWARE

Depending on the application for which they are used, distributed processing systems tend at present to be based on one of two types of processor (in addition to the central mainframe, where one exists), although some specialised products for distributed processing systems have recently been developed.

1. Small general-purpose computers
   Where there is either a significant amount of administrative (as opposed to operational) work, or a mixed load of batch and interactive processing, the distributed processors may well be small general-purpose computers, such as the IBM System/3 and its successors (the System/32, 34, etc.), the ICL 2903 range, or the Burroughs B80. These machines, which could be described as small business systems, will typically have an operating system that is a scaled-down version of a mainframe operating system, and that includes routines that enable it to drive local interactive terminals for data entry or access to data files. Consequently, they tend to require a specialist operator. When communicating with a mainframe or with some other system, they will often emulate a remote batch terminal, and the basis of communication will therefore be batch rather than interactive processing. Non-IBM machines usually emulate an IBM terminal as well as handling their supplier's own protocol.

2. Minicomputers
   By contrast, where the prime purpose of the distributed processor is to drive interactive terminals that are used for operational purposes, then a minicomputer might be used. The Lowe's Accusale system, described on pages 18 to 20, is an example of this type of system. The minicomputer suppliers, such as Digital Equipment Corporation, Data General and Hewlett-Packard, designed their products initially for process control and similar applications. Existing products aimed at the data processing market are generalisations of these initial products, but they retain the sophisticated interrupt structures and other architectural features that make them particularly suited to handling terminals.

Since the early days of minicomputers, the suppliers have also built up their operating software so that it matches the mainframe suppliers' software in sophistication if not yet in range or completeness. Most suppliers, for example, offer compilers for several industry standard high-level languages. Many have database management software. Some, like CMC with their Reality system, offer advanced system building tools. As with the small business systems mentioned above, communication with a mainframe will normally be on a batch basis. Several suppliers now offer networking software, however, and we discuss this later in this chapter.

In summary then, the minicomputer is fast overcoming its reputation as a machine for the "hands on" specialist or the adventurous. At the same time, it offers definite advantages when handling terminals compared with small business systems whose roots are in a batch environment. It should be added, however, that not all the small systems offered by the mainframe suppliers are in the latter category. Honeywell's Level 6, ICL's (formerly Singer's) System Ten and Univac's (formerly Varian's) V-77 all owe more to the minicomputer than to the general-purpose data processing system.

3. Specialised products
   More recently, some products have been introduced that are designed specifically for distributed processing. IBM's 8100, announced in 1978, could be said to have given distributed processing its official seal of respectability. It has since been followed by other products that are intended to be used in a similar way. The 8100 is capable of concurrent autonomous local processing and interactive communication with a remote mainframe system. This means that it can be expected to compete in terms of function with its minicomputer competitors, although it will probably not compete on price. Honeywell's Level 6 system has also been integrated into the supplier's overall scheme, called Distributed Systems Environment, and it has a similar range of capabilities to those of the 8100. The major advantage these products have over their minicomputer competitors is that they are apparently guaranteed to be compatible with the mainframe, because they are purchased from the same source.

FILE MANAGEMENT

Data files accessed by distributed processing systems can be centralised, decentralised, or a combination of both. The reasons for decentralising data files include:

— To provide faster access to the data, because the queues will be smaller, and there will be no communication delays.

— To produce consistent terminal response times, because the load tends to be less variable.

— To provide local control over local data.

— To help reduce data communications costs.

File management problems with individual processing systems are no greater and no less than those with a centralised system. If communication is necessary between files (for example, to capture summary data for corporate use), or from a system to a remote file (because access is not entirely localised), then the same problems are encountered as are encountered with a centralised database, except that they are magnified by the limited control each system has over the others. These problems are particularly associated with file updating, where the processes that update the files need to be synchronised to allow integrity to be maintained. Communication via a transmission link increases both the overhead and the difficulty involved in maintaining synchronisation.

There are also problems of control. Satisfactory communication between files demands not only a reliable method of communicating but also a common understanding of what is being communicated. In other words, it demands a data administration function. (This concept is described in detail in Foundation Report No. 12.) To do his job properly, the data administrator will need to have the means to control both the structure and the content of remote data files, so that he can ensure that a satisfactory level of consistency and coherence is maintained to permit information to be transferred freely throughout the organisation. He may also need some monitoring tools to enable him to police his recommendations.

In an ideal world, database management software would permit organisations to establish a distributed database under centralised administration and control. Regrettably, the technical problems involved (not least of those of synchronisation discussed above) will probably not be fully solved in this decade. In the interim, the software may be so complex as to disqualify itself for many applications where ease of use and simplicity at the local site are pre-requisites. We expect, therefore, that data administration will rely for some time on an ad hoc approach, and that many organisations may need to develop their own tools to support it.

It is not possible to cover fully in this report the complex topic of file design, but we discuss some of the key problems below.

If a system designer has the problem that a collection of data has to be distributed to a number of sites, he has to choose between:

- Replication, where all the data is held at each site.
- Partitioning, where part of the data is held at each site, normally according to the data that the site owns and the site's need to access it.

Those two choices generate their own special problems as discussed below:

1. Replication
   Replication requires that all copies of the information should be kept in synchronisation. The problem is lessened if the replication is not particularly time-critical (as, for example, with price lists) so that data can be collected periodically for updating (perhaps centrally) and then be re-distributed. If this approach cannot be used, complex software mechanisms are necessary to handle updating, and the updating procedure will be at a particular risk during and after a system failure. Replication is therefore most appropriate where retrieval access is a predominant factor. One or two authorities have suggested that replication is feasible only when the level of updating drops below 10% of all accesses.

2. Partitioning
   If files are partitioned, a program that requires a particular item of information has the problem of finding that information. The normal solution would be a system of directories, and ideally, a copy of each directory should be held at each site. (This approach presupposes that the directory itself is not volatile but, if it is, it introduces the problem of updating replicated information discussed above.) A single copy could, however, be more easily held at the centre if the proportion of non-localised accesses permitted it.

   Each system, it is assumed, would know what information was held locally. It may not, of course, always be possible to determine from the directory or the directories exactly where the information is. If, for example, a search is requested on properties that are not recorded in the directory, there will often be no alternative but to search all partitions. Consequently, to be able to design a directory scheme it is necessary to have a detailed knowledge of how the information is likely to be accessed and also how frequently.

   Partitioning complements replication in that it is most suited for handling information that has a high proportion of update access. If a high proportion of the updating is non-localised, however, the advantage is lost. Centralised storage may then be the only way of retaining a reasonable level of control.

COMMUNICATIONS

To some extent, our definition of distributed processing, given in chapter 2, was an over-simplification, since it did not attempt to differentiate between the different types of communications scheme that could be adopted. The techniques for communicating between distributed processors and the centre cover a wide range of sophistication including:

- A straightforward arrangement whereby data is captured locally on cassette tape or some other medium, and is then transferred off-line to the centre at the end of the day.
- A point-to-point communications scheme.
- A complex networking scheme.

As distributed systems develop in number and complexity, and as their interaction with one another and with the centre increases, so the need for a more sophisticated communications structure will increase. The point to bear in mind when deciding on the communications infrastructure for distributed processing is an obvious one. The communications system should reflect the needs of the applications which it supports. One of the great advantages of distributed processing is that it can improve the end-user's interface with computer systems and enhance his control over them. When the communications system is being implemented, care must be taken to ensure that it neither subtracts from nor nullifies that advantage.

At present, most computer suppliers offer their own networking software, or architecture. (The term architecture is generally used to describe the overall scheme or the blueprint, rather than a product that represents a particular realisation of the scheme.) The specialist communications suppliers also offer a limited range of products.

Figure 7 gives a summary of the communications products offered by some of the major mainframe and minicomputer suppliers. These products can be divided into three categories:

- Mainframe-centred products, where overall control of the network and of the devices connected to it is sited in one or more large mainframe systems. In its present form, IBM's Systems Network Architecture is typical of these products.
- Network-centred products, where the devices that handle the communications traffic also provide network management facilities, and often these are distributed throughout the network. Univac's Distributed Communications Architecture and Prime's Primenet are both intended to operate in this way.
- Distributed products, where each system controls its own locally attached devices and communicates across a passive network as and when necessary.

To the extent that it is possible to generalise about each of the above three categories, it can be said that:

- Mainframe-centred products will be most appropriate where the requirements are very diverse, where centralised control is of prime importance and where traffic flow is mainly between the periphery and the centre.

— Network-centred products will be most appropriate where the traffic patterns are more varied and where the network configuration is less stable.

— Distributed products will be most appropriate where most of the processing is local, and there is less reliance on communication to maintain service.

Figure 7 Summary of networking products

Supplier	Architecture (Product) name	Category	Key features
Computer Automation	Virtual Network (SyFa)	N	FS, X25
Data General	(Xodiac)	D	PC, PS, X25
Digital Equipment	Digital Network Architecture (Decnet-20)	D	PC, FS
Hewlett-Packard	Distributed Systems Network (DS-1000, DS-3000)	D	PC, PS, FS
Honeywell	Distributed Systems Environment	M/D	PC, FS
IBM	Systems Network Architecture	M (becoming N?)	PC, DL
Prime	Primenet	N/D	PC, PS, FS, LN, X25
Tandem	Non-stop (Guardian/Expand, Axcess)	D	PS, FS, X25

Category: M — Mainframe-centred
N — Network-centred
D — Distributed

Features: PC — Supports inter-program communication
PS — Supports peripheral sharing
FS — Supports file sharing
LN — Supports high-speed local networks
X25 — Supports the CCITT X25 interface to packet-switching networks
DL — Supports down-line loading of programs

An organisation that is looking for a networking product to support a distributed processing scheme will naturally first look to the major supplier of the computers. Where the major supplier is not immediately obvious, or his product is not suitable, an organisation should use selection criteria that take account of some or all of the factors listed in figure 8.

Figure 8 Selection criteria for distributed processing networks

Application characteristics

Support needed for the communication modes
- Batch data transfer
- Interactive access from terminals
- Access to remote data files
- Access to remote peripherals
- Access to remote programs

Support needed for the data structures
- Replicated or partitioned files
- Directory handling

Equipment characteristics

Support required for the processors that are to be used

Support required for the network structure
- Point-to-point/multipoint
- Hierarchical/ring/mesh

Interfaces to other equipment and services
- Public packet-switched and circuit-switched networks
- In-house private networks
- Existing terminals and processors

Qualities

Growth capability to cater for changing requirements
- Modularity
- Expandability

Support facilities for system development and maintenance
- Down-line loading of programs
- Remote debugging

Support facilities for network management
- Remote diagnostics
- Ease of reconfiguration
- Recovery from failures

Impact on the user interface

Whatever facilities the various networking products offer, they will demand that the user makes a major initial effort to understand how they work, and to configure the software to suit his particular requirements. But once he has made the initial effort, the user is well placed to obtain the following benefits:

— Greater flexibility in configuring the network.

— Easier network management (for example, the remote diagnosis of errors).

— Easier and better co-ordination of communication between systems.

— Better use of equipment resources.

With one or two exceptions, all the products shown in figure 7 are designed primarily for networks of private leased lines. In the US, some users are now adopting the principle of building their own local networks at each site and using public packet-switching services (such as Tymnet or Telenet) to link the sites. As the European public packet-switching services come on-stream, European users will be able to adopt a similar approach. Except in France, where Transpac's tariff has been designed deliberately to attract bulk traffic, the cost of using a public service is likely to be a deterrent at present. But time may well remove this deterrent.

Figure 9 shows the break-even cost for packet-switching against circuit-switching solutions, based on activity and distance between sites. The reducing cost of the processing that is used to switch the packets will move the break-even line towards the left. Also, the build-up of traffic on public packet-switching networks may bring down tariffs. Using a public service has the further advantage of reducing in-house network management activities. But despite the benefits promised by public packet-switching services, some large organisations may well choose to implement their own internal packet-switching network.

Primenet is the best example of a product that has been designed specifically to take advantage of public packet-switching services in the way described. Prime's own corporate network is shown in figure 10 on page 30.

THE IMPACT OF MICROELECTRONICS

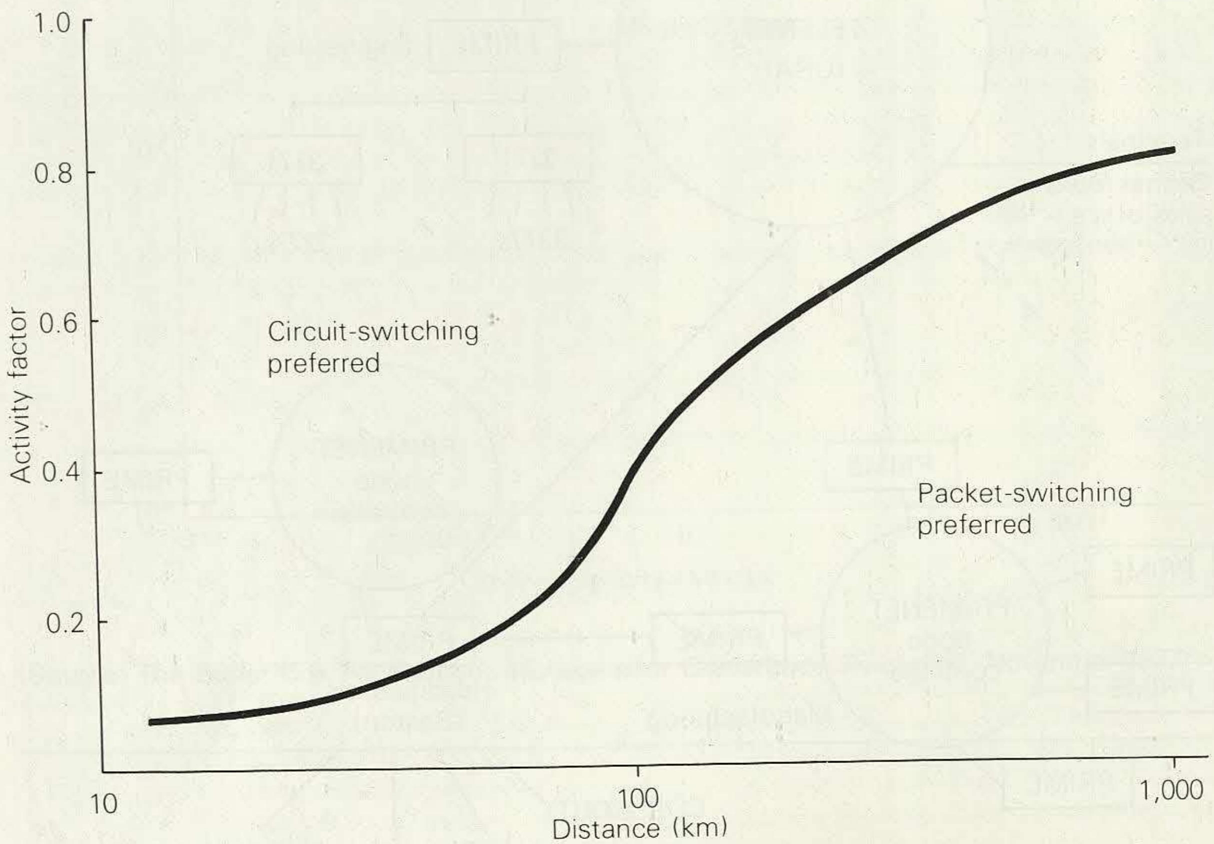
The technological phenomenon underlying distributed processing is microelectronics, which has brought about the rapid decline in the cost of processing. Microelectronics particularly influences the cost of the central processing unit and the main memory, both of which are devices in which solid-state circuitry predominates. Those two devices are the very elements of a centralised configuration that are most likely to be replicated in a decentralised equivalent. Also, for this new memory technology, the graph showing cost against memory size has a much flatter curve than the equivalent graph for older technologies. Consequently, to replicate a large system in many smaller units brings less of a cost penalty, and time will tilt the balance in favour of decentralised solutions.

As levels of integration in microcircuitry increase, so more and more distributed processing applications will become feasible using solid-state devices in all the distributed components. For example, the Japanese are planning to produce a 2 Mbyte dynamic memory chip this year. Solid-state mass storage technology (for example, bubble) is soon expected to out-perform electro-mechanical devices for capacities up to 100 Mbytes, as figure 11 on page 31 shows.

Developments such as this will not only affect costs, they will also improve hardware reliability and simplify maintenance. Overall, then, these developments will ease the problems involved in supporting geographically dispersed computer systems.

The impact of microelectronics could eventually be even more radical. As levels of circuit integration increase, so the semiconductor companies will look around for markets in which they can exploit the increased intelligence of their devices. The area that interests many of them is the area that the French and the EEC have termed "peri-informatics", which includes distributed computing. Semiconductor companies will be determined to create new opportunities in this area as existing markets reach saturation, and they will do this by finding applications that permit large production runs. The resultant products will be so cheap compared with custom-built devices that organisations will find it extremely difficult to prevent their managers from buying them. Those organisations that have established the right infrastructure and the right control mechanism will have the opportunity to exploit this technology other than in a piecemeal way. At that stage, distributed computing will become unstoppable, and this will probably happen in this decade.

Figure 9 Typical break-even cost relationship for circuit-switching and packet-switching networks



(Source: British Post Office)

Figure 10 Primenet — Prime's corporate network

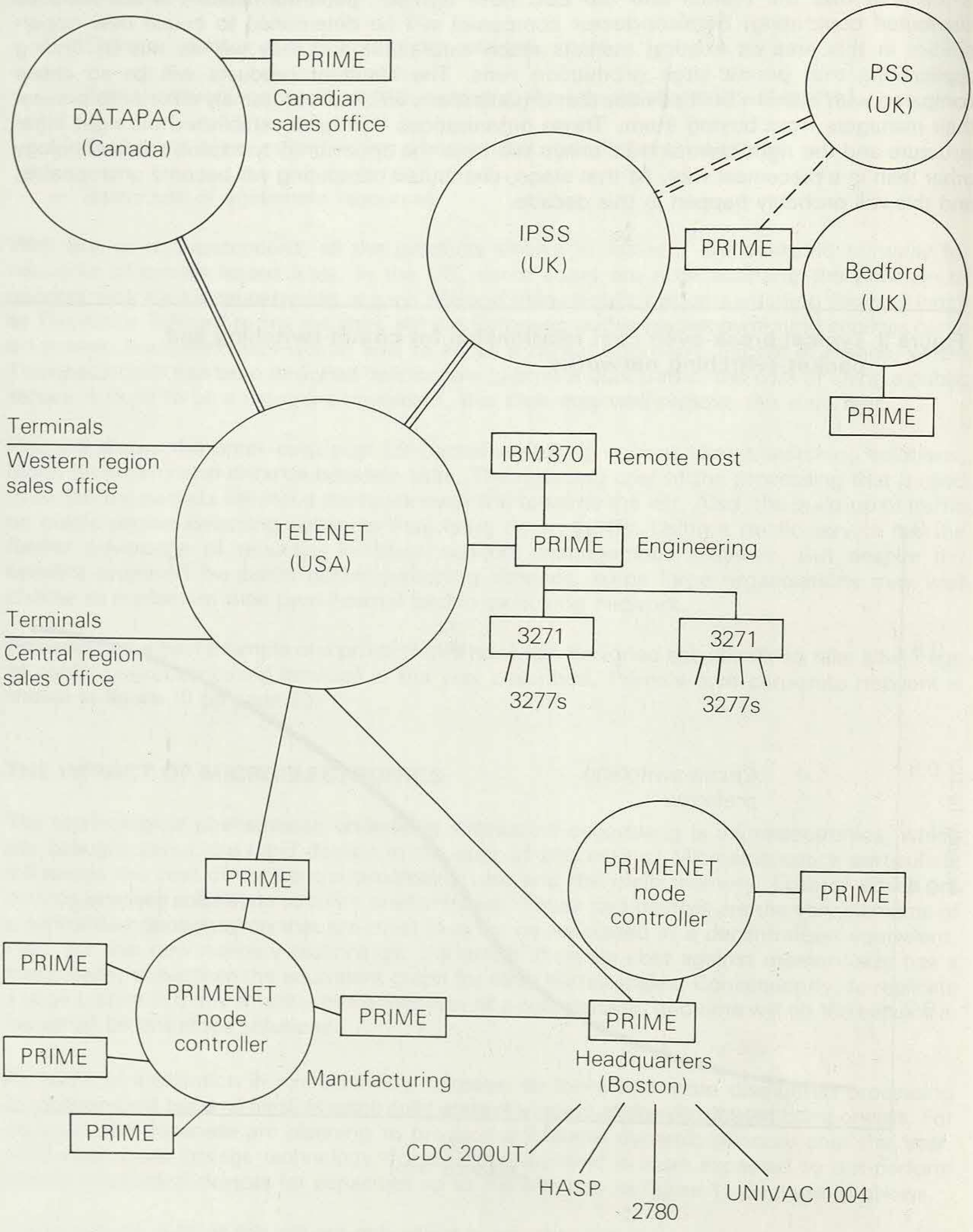
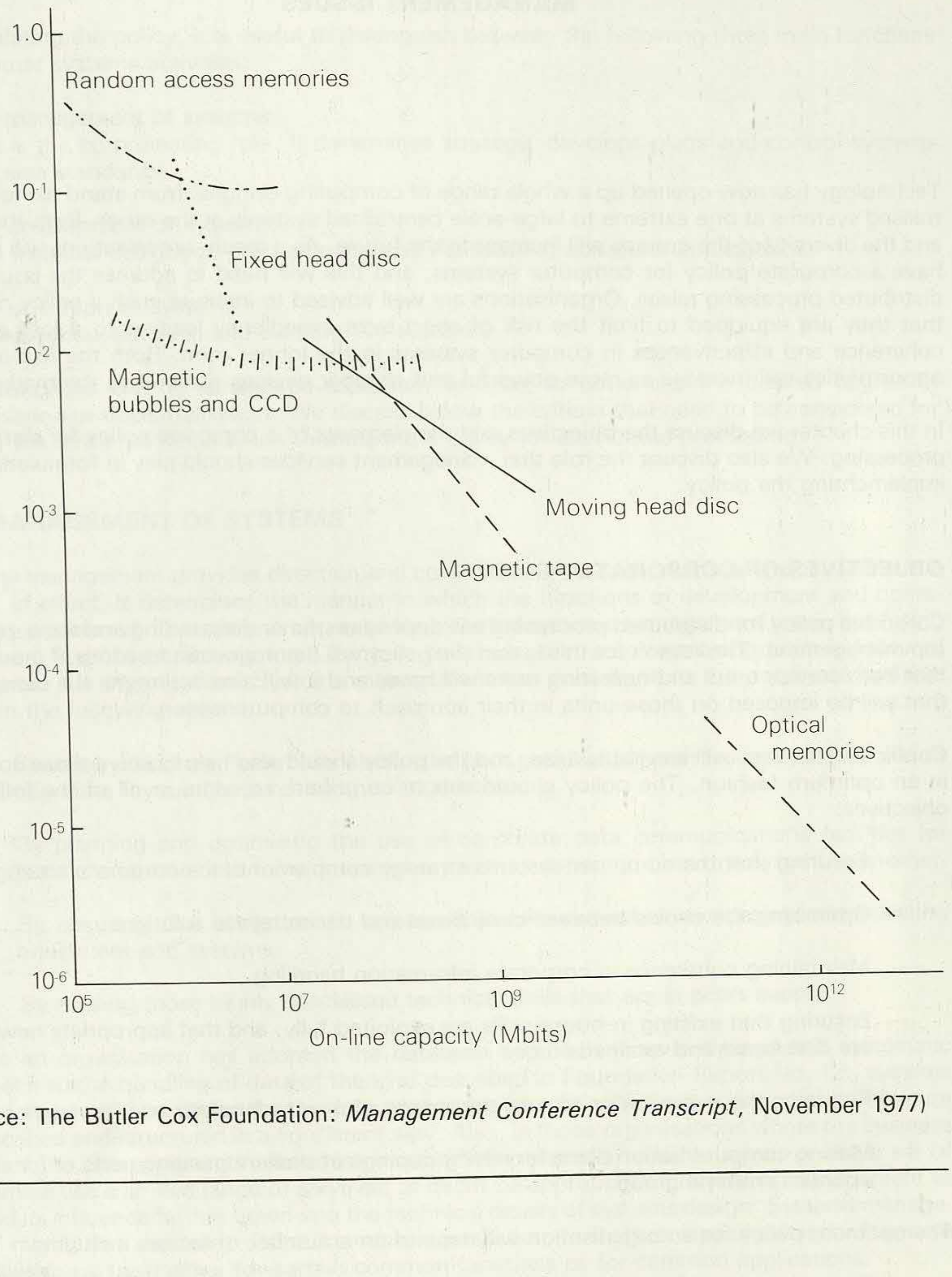


Figure 11 Memory technologies



(Source: The Butler Cox Foundation: *Management Conference Transcript*, November 1977)

CHAPTER 6

MANAGEMENT ISSUES

Technology has now opened up a whole range of computing options, from stand-alone decentralised systems at one extreme to large-scale centralised systems at the other. Both the range and the diversity of the options will increase in the future. As a result, organisations will need to have a corporate policy for computer systems, and this will need to address the issues that distributed processing raises. Organisations are well advised to institute such a policy now, so that they are equipped to limit the risk of short-term expediency leading to a loss of both coherence and effectiveness in computer systems in the longer term. Both the risk and the opportunities will increase as more powerful and cheaper devices come onto the market.

In this chapter we discuss the objectives and the elements of a corporate policy for distributed processing. We also discuss the role that management services should play in formulating and implementing the policy.

OBJECTIVES OF CORPORATE POLICY

Corporate policy for distributed processing will depend on the understanding and the support of top management. The reason for this is that the policy will determine the freedom of movement that both service units and operating units will have, and it will also determine the constraints that will be imposed on those units in their approach to computerisation.

Conflicts of interest will inevitably arise, and the policy should also help to solve those conflicts in an optimum fashion. The policy should aim to contribute to some or all of the following objectives:

- Ensuring that the computer systems strategy complements the corporate strategy.
- Optimising the choice between centralised and decentralised solutions.
- Maintaining coherence in corporate information handling.
- Ensuring that existing in-house skills are exploited fully, and that appropriate new skills are developed and retained.
- Preparing the organisation to take advantage of new technology as it emerges.
- Making computerisation plans for new groupings of similar operating units or for similar activities within a group.

The optimum policy for an organisation will depend on a number of factors including:

- The size of the organisation.
- The management philosophy of the organisation, and in particular the degree of central control that is seen as desirable.

— The geographic distribution of the organisation.

— The nature of the business.

In formulating the policy, it is useful to distinguish between the following three main functions in computer systems activities:

1. The management of systems
   This is the co-ordinating role. It determines strategy, develops plans and control systems and sets standards.

2. The development of systems
   This includes designing, programming and maintaining computer applications.

3. The operation of systems
   This involves installing and running live systems.

These functions can be considered separately in order to determine the desired degree of centralisation or decentralisation. We discuss below the criteria that need to be considered for each of the three functions when formulating a policy for distributed processing.

THE MANAGEMENT OF SYSTEMS

Systems management provides direction and co-ordination for computer systems activities as a whole. In effect, it determines the manner in which the functions of development and operations are performed. There is a strong case for a central systems management team, but failing that, there need to be management controls to ensure that a coherent approach is followed throughout the organisation. Systems management can contribute to the corporate systems effort in the following ways:

— By using corporate "clout" to obtain the best possible support from suppliers, and to negotiate favourable purchasing arrangements.

— By planning and optimising the use of corporate data communications facilities for transferring data both from the periphery to the centre and from one system to another.

— By ensuring that a systematic approach is followed when evaluating and selecting equipment and systems.

— By pooling those highly specialised technical skills that are in short supply.

Where an organisation has adopted the database approach (a co-ordinated and systematic approach to the handling of data of the kind described in Foundation Report No. 12), systems management will also undertake the data administration role, which will ensure that information is described and structured in a consistent way. Also, in those organisations where the business areas of the operating units are closely coupled (for example, a bank, where a single set of customers use a limited range of services), it might be appropriate for systems management to extend its influence further down into the technical details of systems design. Systems management might, for example, define standard transaction formats or specify standard system modules to be used either for certain common functions or for common applications.

To be effective, the systems management function must have the necessary authority. The sanctions that are used and the way in which they are applied depend on current practices and current management philosophy. However, as devices become cheaper, capital expenditure sanctions alone are unlikely to be effective. The only certain way in which the systems management team will carry authority effectively will be by earning respect through performing their tasks successfully.

THE DEVELOPMENT OF SYSTEMS

It is more difficult to determine the right approach to systems development than it is to determine it for systems management, because the latter is, in some respects, a natural monopoly. It will often be necessary to apply criteria on either a per unit or a per application basis. For example, an organisation may choose to centralise the development of all personnel applications and financial modelling applications, and to decentralise the development of all other applications.

The advantages of centralised systems development are as follows:

- It eases the problem of coping with peaks and troughs in demand.
- It is the easiest way to ensure that inter-relationships between systems are both catered for and exploited.
- It is probably a better way of getting value out of the scarce resources of systems analysts and programmers.
- It makes it easier to enforce professional standards.
- It provides a career path for data processing staff and particularly for those who have advanced technical skills (for example, in database or data communications).

On the other hand, the following advantages are claimed for the decentralisation of systems development.

- It supports a management philosophy of decentralised authority.
- It improves communications between systems development staff and end-user staff.
- It shortens the system development cycle for key applications.
- It enables the user manager to set his own priorities both for new developments and maintenance work.
- It is the best way of teaching user managers about the realities of computer systems development.

In summary then, centralisation provides benefits as regards personnel and the quality of systems, whereas decentralisation provides mainly management benefits.

THE OPERATION OF SYSTEMS

The decentralisation of operations implies, of course, that both the computer systems and the staff that operate them are dispersed. As with systems development, it may be best to apply criteria selectively. Selection will not only be by operating unit or application, but also by function within an application. In other words, it will be necessary to apply design criteria when determining which functions and which data elements for a particular application are to be centralised, and which are to be decentralised. The essence of a distributed approach, as opposed to either a fully decentralised or a fully centralised approach, is that it permits this kind of choice to be made. Methods that can be used to determine the best choice are discussed on page 39.

We set out below the respective advantages of centralised and decentralised operations. An organisation needs to make a comparative evaluation of those advantages before it decides its overall policy.

The advantages of centralised operations are:

- Economies of scale, which still apply for some devices (such as very large mass storage subsystems or high-speed printers) and for some applications (such as financial modelling or complex database systems). For effective processing power alone, though, the case for economy of scale is now arguable.
- The capacity to withstand rapid growth (although not necessarily without difficulty and heavy expenditure).
- Better service and support from suppliers.
- Easier internal controls and higher levels of physical security over access to equipment.
- Better security features in operating software.
- The availability of a wider range of applications software and utility software.
- Easier training of, and greater flexibility in the development of operations staff.
- A limited need for user managers to learn about installing and operating computers.

Decentralised operations have the following advantages:

- A reduced vulnerability of systems to hardware failure, or sabotage or industrial action.
- Better performance and greater flexibility with terminal-based applications.
- The facility to upgrade equipment progressively (although the equipment can be vulnerable to rapid growth, because of the more rigid limitations on capacity).
- The ability to meet end-users' desires to own their systems and data, and to control their operational priorities.
- Reduced equipment operating overheads (although there are substantial risks).

Centralised operations therefore still have much to commend them. On the other hand, although there may be fewer reasons to adopt a decentralised approach, the reasons that can be advanced in its favour are extremely powerful and can become irresistible. Even though many organisations are likely to persevere with a substantial effort in systems operating centrally, we believe that few will be able to ignore the need for some decentralisation of operations. If this decentralisation is to take place on any scale, then the problems of designing, developing and supporting distributed processing systems will need to be tackled.

THE ROLE OF MANAGEMENT SERVICES DEPARTMENTS

Management services departments are generally charged with preserving an existing massive corporate investment in centralised systems, but they are also expected to exploit any new technology that is developed. These two tasks confront management services departments with at best a trying time and at worst with an impossible job. They are likely to have to deal with unfair comparisons that are made between working systems that were designed for, and that operate with, yesterday's technology, and extremely cheap and apparently easy-to-use new options that claim to provide tomorrow's technology today. Faced with pressures to decentralise responsibility for the development, the operation, and sometimes even the management of computer systems activities, management services departments will be legitimately concerned that:

- Users will repeat all the mistakes data processing people made in the 1960s, and these may result in unmaintainable systems and, eventually, chaos.

- Systems interfaces will not be controlled adequately, so that corporate control systems will become unreliable.

- Career paths, particularly for some senior staff, will be damaged or even destroyed, and some jobs will be lost.

- Newcomers (especially external systems houses) will come in and implement quick and inferior solutions and will receive praise for their work. They will then leave behind a mess for management services to clear up.

- Hardware and software will be procured by inadequately skilled staff in an uncoordinated way.

- The load on the existing equipment will be allowed to run down too quickly, leaving management services with an expensive white elephant to dispose of.

None of these concerns are, we believe, arguments against decentralisation, but they are arguments in favour of managing the transition properly as it occurs. The technical concerns need to be addressed by the systems management function that we have outlined, and the career worries need to be addressed by appropriate recruitment policies, and possibly by a policy of encouraging some staff to pursue promotion paths in user departments, rather than in management services. We say this not because we believe that these changes can be achieved as easily as they can be written, but rather because we believe that the pressures to decentralise operations and, sometimes, to decentralise development, will eventually become almost impossible to resist. The danger is that if a management services department resists these pressures too vigorously, it will lose credibility, and with it the ability to assume the vital managing and coordinating role that it is best placed to fulfil.

The position of management services departments who now preside over a centralised data processing service can be compared with that of computer service bureaux who were faced with the threat that minicomputers made to their traditional business (which was also centralised on a large computer). The more far-sighted bureaux have countered the threat by offering to install minicomputer systems for their customers. By ensuring that those systems were compatible with their centralised bureau systems, they have, as a consequence, generated new business from developing the programs that are run on their customers' minicomputers, from handling peak loads and from providing stand-by facilities. This additional business has replaced the losses in their traditional business.

Management services managers should therefore seek:

— To ensure that a balanced corporate policy for computer systems is adopted.

— To retain and to build up the vital systems management skills in equipment selection, in data communications, in distributed systems architecture and in data analysis.

— To obtain or to consolidate control over the infrastructure on which systems are based (the communications network, the data dictionary and the standards).

Finally, we believe that the management services department, if it has not already done so, needs urgently to gain experience in implementing distributed processing solutions, even if this means diluting its existing skills or moving into unfamiliar and risky territory. There are still uncertainties in some technical areas (for example, in networking and distributed database), but there is unlikely to be a marked improvement in these for some time. Delay, therefore, only increases the risk that the initiative will slip away from the management services department altogether.

OPTIONS FOR THE DECENTRALISATION OF STAFF

Any discussion about a policy for distributed processing is incomplete unless it also considers the practical options that are available for implementing that policy. We start from the reasonable assumption that those organisations that wish to gain the benefits of distributed processing will wish to do so without losing the existing valuable advantages of centralisation, and without unnecessarily dislocating the organisation or the staff.

Systems management is, as we have said, something of a natural monopoly, and so it should be centralised. Decentralised operations present no problems that cannot now be avoided by good design, or that cannot be handled by sound operating practice or appropriate support arrangements. Those problems may be new to some organisations, but experience shows that they can be solved without undue effort. This leaves system development as the major problem area because, there, decentralisation affords both important benefits and significant risks.

Two different approaches to the decentralisation of system development have been adopted with some degree of success. The first and most common approach has been to decentralise systems and programming work, but to retain a central pool of staff who have specialised skills and experience. These centralised staff then operate in a consulting role, and their contribution to user projects has included:

— Helping to determine the optimum system structure that balances the advantages of local autonomy against the need for centralised back-up and support.

— Getting the project started by using either prototyping tools or timesharing services.

— Advising on and approving equipment selection.

— Helping in specialist technical areas, such as database and communications.

— Providing a project manager.

The second, and less common approach has been the task force approach. One large European bank set up a task force consisting of internal DP staff and some of the major supplier's staff to implement its first major distributed processing application. The task force was given almost a turnkey mandate to implement the system, and it reported to a committee of senior management. This arrangement distanced the task force sufficiently far from the centralised DP operation to permit it to secure resources as and when it needed them (the DP department was overstretched anyway), and also to cushion the task force against the cultural shock of a new approach to computing.

This task force approach has great attractions as a means of coping with the first distributed processing project, but conceivably it could also be adopted on a wider basis than that. The centre would thus provide the operating companies with turnkey implementation and facilities management functions.

We believe that some of the important benefits of distributed processing are obtained only if systems development is decentralised. But some organisations may quite legitimately wish to forgo those benefits in the interests of tight central control.

EQUIPMENT POLICY

To contain the complications that arise out of distributed processing, an organisation needs to have an equipment policy. The three main options (which we discuss below) appear to be:

- Standardising on a single supplier.
- Selecting from an approved list of suppliers.
- Adopting a set of international or industry standards.

It is also possible to combine the above three options, or certain facets of them, in different ways.

Depending on an organisation's viewpoint, it may wish to prohibit the purchases of any devices that do not conform to its equipment policy. Alternatively, it may use the sanction of limiting the support that is provided from within the organisation.

1. Standardising on a single supplier
   This is the safest but also the most limiting approach, even where the approved supplier is IBM. The plain fact is that the market is already too wide (and is rapidly becoming wider) for any one supplier, however powerful, to provide comprehensive coverage. Specialist suppliers will increasingly offer very attractive products that are aimed at particular sections of the market. A too restrictive policy could therefore leave an organisation at a disadvantage compared with its more adventurous competitors.

   In the field of distributed processing in particular, IBM's belated conversion to the concept has left it some way behind the major minicomputer companies, and also some way behind some of its mainframe competitors, particularly in software provision. Nonetheless, no other supplier can match IBM's range of available hardware and software products. Standardising on the products of a single supplier does not, however, necessarily guarantee full compatibility. For example, the operating systems on IBM's large (303X), medium (4300) and distributed (8100) processors are all different.

2. Selecting from an approved list of suppliers
   This option overcomes some of the objections to the single supplier approach, but it obviously involves the use of greater skills to support diverse equipment and to achieve the level of compatibility necessary for interworking. The equipment market will become increasingly competitive over the next few years, and so an organisation that selects this option will need to keep the approved list under constant review.

3. Adopting a set of international or industry standards
   This option probably provides the way an organisation can get closest to a policy that permits it to have the widest possible choice of equipment whilst retaining the required degree of control. But of the three options, it has the highest potential risk, and it places the greatest technical demands on those staff that have to implement the policy. It will never be wise for two reasons to rely on suppliers' assurances that their products meet the specified standards. Firstly, the standards themselves may be ambiguous or inadequate and secondly, suppliers may interpret those standards in an optimistic way. The staff concerned will therefore need to establish a detailed understanding of the standards for themselves, and they will possibly need to devise test aids to verify suppliers' claims. It will, of course, be necessary to constantly review both the choice and the composition of standards in order to keep the policy up to date.

Flexibility can be introduced into this option by relating the restrictiveness of the standards that must be complied with to the level at which interworking is likely to take place. Thus, there would be no restriction on acquiring stand-alone devices, but those devices that communicate in off-line batch mode or in real-time mode, or that share data or programs, would be subject to progressively more rigorous rules.

EVALUATING ALTERNATIVE SYSTEM STRUCTURES

When an organisation is determining the optimum structure for a particular system a formal methodology is a great advantage, because it can be used to counter suspicions that centralised DP staff are salesmen in disguise for the central computer. A structure for such a methodology has been put forward by a group working at Kansas State University (J. Slonim, et al., *Information Management*, Vol. 2, No. 1, 1979). This consists of defining the criteria for evaluating a distributed system (and some of those the group put forward are listed in figure 12), assigning a weight to each, and then giving each major system option (centralised, decentralised with partitioned files, decentralised with replicated files, etc.) a rating for each evaluation criterion. Those criteria that are relevant for a particular application can then be identified, and they can then be used to calculate an evaluation index for each system option. The evaluation criteria and the ratings could initially be based on generally held views, and could progressively be refined to reflect an organisation's particular experience and priorities.

An alternative approach for deciding whether to use centralised or decentralised equipment has been proposed in a draft paper prepared by the Centre of Information Systems Research at the Sloan School of Management, Massachusetts Institute of Technology (principal author John F. Rockart, and yet to be published). The paper argues that the decision on whether to use decentralised or centralised equipment should be based on logical application groups (LAGs). Each LAG would be concerned with a logically separate process performed by the organisation. Consequently, the transfer of data within a LAG would be intensive, but the transfer of data between LAGs would be limited and well defined. Examples of LAGs are order entry applications (which might be decentralised) and financial accounting applications (which might be centralised). Seabrook

In many cases, it will be possible to adopt the same approach for all operating units that use a particular LAG. Sometimes, however, it may be necessary to make a separate decision for each operating unit (for example, where there are considerable differences in the sizes of the operating units).

An important aspect of the systems management function will be to help formulate such decisions about the structure of computer systems. Correct (or incorrect) decisions about the structure will probably have a longer-lasting impact on the organisation than will decisions about the equipment supplier. Decisions about the structure of computer systems therefore need to be accorded the attention and the care that they clearly demand.

Figure 12 Selected evaluation criteria for distributed system options

Operational characteristics
Locality of the data
Conformity to data standards
Machine independence
Evolutionary growth of the data
Diverse requirements of users
Security

System performance
System complexity
Program compatibility
Data redundancy
System throughput
Response time overhead

Economics
Personnel cost
Hardware cost
Data communication cost
Expansion cost
Update cost (per unit)
Retrieval cost (per unit)

Update and retrieval characteristics
Update from different location
Deadlock
Batch query
On-line query
Unrestricted query

Database size

User charateristics
Number of local users
Number of distributed users

(Source: J. Slonim, et al.)

CHAPTER 7

CONCLUSION

Peter Drucker defines one of the tasks of management as being "to organise work according to its own logic". Distributed computing is a means of bringing computer systems much more closely into line with that requirement. It is in this respect that it differs from earlier fashions in computing, such as the ill-fated Integrated Management Information Systems. A distributed computing system does not attempt to impose on the people who use it a way of working that suits the currently available technology. Rather it does the reverse, because those cost barriers that have previously restricted users' computing options have now fallen sufficiently for computer systems to correspond more and more closely with the management needs, the practical needs and the psychological needs of those who use them.

Distributed systems and decentralised systems can now handle certain applications more effectively than can centralised systems. Technology and market forces will increase the attractiveness of these solutions to the point where, probably in the second half of this decade, the growth of distributed computing will become irresistible (though there will still be a place for centralised computing). Consequently, as soon as their current data processing capabilities permit them to do so, organisations need to encourage their users to run their own computer systems under properly controlled conditions. We have outlined in this report the policy issues that we believe need to be addressed as a preliminary to this.

Finally, there can be little doubt that office automation applications, if they are to meet their full potential, will demand a policy framework that is similar to the one required for distributed processing. Those applications are likely to share much of the same equipment as distributed processing systems, and they will often have the same requirements for local autonomy and central co-ordination. In more than one sense, therefore, distributed processing is the right route for an organisation's computing policy to take in the 1980s.

Photoset and printed in England by Flexiprint Ltd, Worthing, Sussex

The Butler Cox Foundation

Report Series No 18 Distributed Processing: Management Issues

Please record your interest in follow-up work on the subject of this report, highlighting those aspects of the topic of most interest to your organisation.

Completed by ....

Job title ....

Organisation ....

Address ....

....

Please return to: The Butler Cox Foundation
Butler Cox & Partners Limited
Morley House
26 Holborn Viaduct
London EC1A 2BP

The Butler Cox Foundation

Report Series No 18 Distributed Processing: Management Issues

Please note below any points on which you would like the author to expand at the professional and technical seminar associated with this report.

In order to take account of your comments we need to receive them from you by 27 June 1980.

Completed by . . . .

Job title . . . .

Organisation . . . .

Address . . . .

. . . .

Please return to: The Butler Cox Foundation
Butler Cox & Partners Limited
Morley House
26 Holborn Viaduct
London EC1A 2BP

The Butler Cox Foundation

Butler Cox & Partners Limited
Morley House, 26-30 Holborn Viaduct, London EC1A 2BP
Tel 01-583 9381, Telex 8813717-GARFLD

SISDOCONSULT
20123 Milano – Via Caradosso 7 – Italy
Tel 86.53.55/87.62.27

Akzo Systems B.V.
Velperweg 76, Arnhem, The Netherlands
Tel 85-662629

Butler Cox & Partners Limited
216 Cooper Center, Pennsauken, New Jersey 08109, USA
Tel (609) 665 3210

Printed in England