Gaines, B. R. (1978). Minicomputers in business applications in the next decade. In Infotech State of Art Report on "Minis Versus Mainframes", pp. 51–81. Infotech International, Berkshire, UK.

MINICOMPUTERS IN BUSINESS APPLICATIONS IN THE NEXT DECADE

B R Gaines

Man-Machine Systems Laboratory
Department of Electrical Engineering Science
University of Essex
Colchester Essex

ACKNOWLEDGEMENTS

The contents of this paper owe much to discussions over the years with many colleagues. I wish particularly to thank Bruce Anderson, Frank Coakley, Chris Corbett, Walter Doherty, Peter Facey, John Gedye, David Hill, Mike Hills, Tim Kennedy, Sandy McKellar, Peter Madams, Neville Moss, John Sams, Alec Trustram Eve, Peter White and Ian Witten, for lively discussions.

© B R Gaines 1978

B R GAINES is Chairman of the Department of Electrical Engineering Science at the University of Essex. He has been Technical Director of a timesharing service company a minicomputer manufacturing company in Great Britain, and consultant on computer engineering to a number of British and American companies. His commercial experience includes the design of general purpose and special purpose computer hardware, and the development of operating systems, languages and a wide variety of application packages. Professor Gaines' main area of research has been on human factors in computing, particularly on effective dialogue programming for interaction with naive users.

MINICOMPUTERS IN BUSINESS APPLICATIONS IN THE NEXT DECADE

INTRODUCTION

This paper is based on some ten years experience of designing and implementing a wide range of interactive systems based on minicomputers. Applications have included industrial and clinical instrumentation, psychological testing, market research, hospital administration, and a variety of commercial systems for security and foreign exchange dealing and banking. Previous reports have detailed the applications (011,018,022,023), given details of the interactive language used (020) and its implementation (015), and analysed the problems of programming interactive dialogue for naive users (018,019).

This paper attempts to draw out the main lessons learnt from these applications, particularly the role of minicomputers themselves in business applications, and to project the experience gained into the next decade. It takes a variety of points of view, but makes the main future projections in terms of:

- *Hardware*: processor, backing stores, communications and terminals
- *Software*: operating systems, network protocols, interactive languages, and applications packages
- *Peopleware*: developmental techniques, programming dialogue, and social factors in computer introduction.

However, this paper is not intended to be a 'technological forecast' in the usual sense of the word; books such as (050,009,049) may be consulted for projections of the growth of computer technologies and associated usage, and I have already given some general projections of this nature (012,017). Rather, it attempts to fill in some of the background and detail to the more structural and organizational forecasts that have been made of the future roles of computers in business life — clothing the skeleton of such general projections as Nolan's 'stage' hypotheses (038,042). The commercial use of computers, particularly the 'routine' (and most significant) usage, lags behind technological growth, and we are only now seeing the full exploitation of developments from five to ten years ago. It is the technology that we can see today that will form the basis of the business systems of tomorrow. This paper is concerned with how all the known technologies interact to create both the opportunities and the problems of tomorrow.

MINIS VERSUS MAINFRAMES

The key technology underlying the applications experience on which this paper has been based is the *low cost, small-size, simply-operated minicomputer*, supporting *interactive*

services, *dedicated* to, and *controlled* by, its local users. Since the minicomputer is historically associated with industrial and scientific usage, and the commercial organizations using them would normally be expected to utilize mainframe EDP machines, this experience does contribute to the 'mini-versus-mainframe' debate. However, it is worthwhile making some specific warnings about too simplistic application of this distinction.

The now classic debate of 'minis versus mainframes' catches some significant differences in technology and marketing of these two ill-defined classes. However, it fails to catch many others and, now that it has been joined by the 'midi-mini' and 'mini-micro' debates, there need to be far clearer and more cogent distinctions drawn. I have seen many mini-based systems that lose key advantages of the technology, e g, they need air-conditioning, skilled operator attention, or do not offer fast interactive access. I have also seen some mainframes that offer many advantages that one might wish to claim for e g, the Xerox Laboratories in-house-built PDP-10 at Palo Alto that sits alone in a basement near the pool room having no operator or central lineprinter, being continuously on-line for months at a time, and offering a fast-response reliable service geared closely to the needs of its local user community.

The distinction between minis and mainframes is not clear-cut because it is multi-variate and also involves non-technological factors. Processors themselves nowadays form a continuous spectrum from the lowliest 1-bit, 10 microsecond micro, through the 16-bit, 1 microsecond mini, to the giant pipelined 100-bit, 50 nanosecond array machine. Within that spectrum the mini and EDP mainframe processors sitting either side of the centre look far more similar to one another than they do to either extreme. In terms of peripherals, too, there are systems sensibly using minis, or even micros, with bigger backing stores than on some mainframe configurations. In terms of software, some mini operating systems and language compilers are more flexible, efficient and powerful than those on some mainframe systems.

However, there have been at least fairly clear historic distinctions between minis and mainframes in their marketing, support, expected applications, and so on. Mainframe systems were developed largely for the EDP market and minis for laboratory/industrial systems. A *mainframe* tends to be used as a central system providing a range of general services to diverse users requiring fairly large-scale data processing. The *mini* tends to be used as a local system providing specific facilities on a dedicated basis to a coherent group of users requiring rapid access to small-scale computations on a simple data base. Historically the minicomputer is associated with lower cost systems and hence with lack of software support, particularly operating systems and full-scale, high-level languages since these have tended to be expensive both to develop and to run (in terms of configuration requirements).

These already imprecise distinctions are continually further blurred by rapid changes in hardware, software and applications technologies. However, underlying these surface distinctions there is a more meaningful structure of greater constancy that has little to do with computers as such, but far more to do with the 'model of the firm' those psychological, social and organizational factors that govern the operation of any institution. Questions, such as 'centralization versus decentralization', 'hierarchies versus heterarchies', 'participation versus control', and so on, are all classics in their own right. However, the type of computer operation implemented can interact with and greatly affect these more fundamental issues.

The simple-minded approach of associating mainframes with centralized hierarchical organizations exerting detailed control over their human elements, and distributed mini

systems with decentralized, heterarchical systems allowing employees a high degree of autonomy and participation in local systems design, may seem to fit many organizations, but is itself only a rough approximation to a far more complex picture. The technology makes easier, rather than enforces, one structure or the other, and actual organizations themselves are never the extremes so readily caricatured. What one has to develop both to comprehend and to design effective systems is a detailed model of how these many historic, technological, organizational, etc factors actually operate together.

In the remainder of this paper I shall use the terms 'mini' and 'mainframe' to indicate the population stereotypes outlined above, but the complexity of the underlying distinction should be borne in mind and will be noted again at various points in the discussion. The importance of the many issues underlying the distinction is of course that the most straightforward approach to the design of cost-effective systems is one which matches the application requirements to the total underlying technologies — hardware, software and human. Whilst the skilled system designer and implementor can undoubtedly make silk purses out of sows' ears, those with less skill or less time and effort to waste on designing around unnecessary problems will be looking for the easiest, most robust system design that fits the task in hand. Our prime concern thereafter is to avoid making sows' ears out of what should have been silk purses!

WHY MINICOMPUTERS?

We have noted previously that there are many business applications of computers that fit far better the terminology and techniques of industrial control than they do those of conventional data processing. For instance, a dealer in any commodity whose price is continually fluctuating, be it money, stocks or physical commodities, requires *real-time* information about the deals of his colleagues and the state of the market; he requires *on-stream* processing of this continually changing information against an *on-line* file of historic data about the market.

The dealing situation has indeed proved an attractive one in which to use minicomputers originally designed for industrial process applications (022,023). However, there are also rather more general business requirements which make the use of small, local, interactive minicomputer systems (rather than batch-processing or access to a central utility) very attractive. Whilst one may discuss and justify this in terms of a wide variety of phenomena, some of which have been mentioned in the previous section, there appears to be one single underlying factor and that is *responsiveness*. Responsiveness, the way in which the computer-based system responds to the users' requirements, is the key parameter by which all others can be evaluated. In particular, the initial use of batch-processing centralized mainframes, the swing towards interactive teleprocessing, and the more recent developments in local minicomputers may all be viewed as the result of optimizing system responsiveness given the technology of the time. This 'human factors' aspect of computing systems does have many direct technical implications (012,013, 014) and its influence in the current context is worth pursuing in some depth.

Human factors in system design

Two remarks seem to me to epitomize the role of human factors in computer system design: John Cleary at EUROCOMP '74 in discussing the role of such new system concepts as the 'virtual machine' said:

> "The only virtual machine of interest to a user ... [is] ... one that has abolished all lower levels of system and presents an *understandable* and *sympathetic* face to its user"

and Walter Doherty, Manager of the in-house systems at IBM's Thomas J Watson Research Laboratories, remarked at a seminar at Essex University in September 1977:

> "The computer is a tool to extend man's memory and reasoning power — these are the key bases on which technical advances should be evaluated".

Doherty's remark is a salutary reminder of the real priorities when we are tempted to compare minis and mainframes in purely technical terms: speed, size, reliability, etc, are secondary variables, albeit very important ones, and their effects have to be evaluated in terms of their impact on the primary requirement of providing a cost-effective service. Doherty also presents effectiveness itself in human terms, not of data processing throughput, but as the extension of human memory and reasoning. Again, 'number-crunching' is an important secondary variable but it is not a primary one — we have to consider *why* we are actually doing it (e g, in a batch-processing environment we may perform massive computations just to provide a 'data base' on-line printer paper for a person to 'interact' with later!).

Cleary's remarks about an 'understandable' and 'sympathetic' computer system may at first sight seem emotive terms best suited to sales literature. In fact, however, they have very direct technical interpretations in terms of designing man-computer dialogue. Making it easy for the user to model the system and making the system support and help the user are key aspects of dialogue programming (018). However, in the present context, it is more important to note the change in attitudes to, and roles of, the computer as envisioned in Cleary's remarks. Sympathy and understanding are traits that we might hope for in people, and in requiring them in computer systems we are clearly beginning to accept the computer as a 'colleague' rather than a 'tool'. Doherty's extension of the memory and reasoning power could be applied to passive objects such as books, and to mindless aids such as hand calculators, but we are now also considering a more equal partnership with an active 'thoughtful' colleague, rather than a passive 'mindless' tool.

Clearly this swing towards partnership has a long way to go before real equality is attained (I will not say by whom — many of us feel servants of our systems already!), but the tendency is very apparent in such analyses as Withington's (052) of 'five generations' of computers: from the 'gee whizz' status symbols of the fifties, through the 'paper pushers' of the early sixties, the 'communicators' of the late sixties and early seventies, to the 'information custodians' of the seventies and eighties, and the projected 'action aids' of the eighties and nineties. Another way of viewing the same spectrum of change is to note that the early stages could well be said to form a 'clerical revolution' whereas the later stages show the beginning of a 'managerial revolution'. We gladly gave over the soul-destroying activities of an army of clerks to the computer in the sixties. Now we are moving into an era of shared managerial responsibilities where partnership is both possible and desired. The beginning of the 'man-computer symbiosis' that was conjectured by Licklider (034) in the late fifties is now upon us and will have increasing impact.

As I suggested earlier this variation in the role of computers may also be seen as a change in the responsiveness of computer-based systems. Mean-times-between-failures (MTBFs) of less than 30 minutes up to a few hours were prevalent for business data-processing configurations up to the mid-sixties and the key parameter affecting overall

responsiveness was maintainability. It made sense to centralize the computing service so that maintenance engineers could be continually at hand to deal with the irregular but frequent and expected faults. There was no question of being able to offer an on-line, real-time service, and batch-processing was the most effective technique for maximizing responsiveness by making full use of any processing power as it became available. Thus systems had to be essentially 'hands-off' and responsiveness was a function of turnaround time. Precisely when his work was done did not matter to the user, provided the results were back within a maximum time period.

Clearly, even under these conditions it is not just the response-time that determines responsiveness but also the software aids available. It may be stretching Cleary's terms somewhat, but variations in the 'understandibility' of COBOL compiler diagnostics, and the 'sympathy' with which errors are treated are very apparent even in a batch environment. Clearly the massive granularity of batch-mode interaction places severe limitations on the effectiveness of attempts to aid understanding and be sympathetic. The situation is what Lewis and Cook (033) have termed one of 'telling', where one party to a 'conversation' is only a remote recipient, unable to take an active part.

The advent of machines using solid state devices of increasing reliability and complexity in the late sixties led to a dramatic increase in MTBFs to hundreds and then thousands of hours. The use of computers in an on-line mode became both practically and economically feasible in more than limited experiments. However, the software reliability problems of multi-access timesharing and transaction-processing operating systems had effects similar to the earlier hardware reliability problems. Skilled software maintenance staff were required close at hand and system operation required a substantial nucleus of trained staff attendent on the machine. For many years the software available to interactive users was based on that previously available to batch users and took very little advantage of the closer communication available with the user. Many systems today still operate in what can best be described as 'rapid turnaround batch-processing', a quantitative improvement in response-time leading to a qualitative improvement in responsiveness, but still nothing like that of truly interactive communication — in Lewis and Cook's terminology it is just being 'told' more often!

This, to a large extent, represents the situation up to recent years. Although, cost-effective, operator-free operation of a large timesharing computer utility is possible and in routine use at a number of centres, and interactive software has been developed for many systems that provides a range of dialogue-based services aiding all users from the most naive to the most skilled, most computer services are still run in a mixture of batch, and interactive, fast turnaround batch, modes. However, the continuing growth in solid-state circuit complexity coupled with decreasing costs, size and power consumption, coupled with similar phenomena in essential peripherals such as backing stores, has brought the minicomputer in as a major competitor to the timeshared utility in offering a responsive system.

If the impact of this growing power of minicomputers was purely technical it would be possible to treat it as part of a continuous process of increasing system reliability leading to increased responsiveness. However, there is also a massive social effect: the decreasing cost of computer systems allows end users to buy them on local budgets; the decreasing environment requirements of minicomputers enables end users to house them locally and treat them as normal office equipment; the increasing responsiveness of minicomputer software enables end users to program and use them without recourse to experts; the increasing power of minicomputers enables end users to do most of their work with them. The overall qualitative effect is that end users are becoming de-professionalized and very much less dependent on the day-to-day services of computing

specialists.

To some extent this change was already beginning to occur with timeshared systems where remote users were, initially at least, free of central supervision. However, use of a large central machine is still expensive: there is a substantial overhead to pay for just being connected and there is the possibility of tying up large-scale resources. So that there has been a tendency to build tighter resource-control systems that centrally monitor and control even remote users. The freedom that a local system gives, and the absolute sense of data security attained, are probably more significant than any other factors in making local minicomputer-based systems attractive – these are the strongest 'human factors' of all!

Technical aspects of minicomputer responsiveness

Although I have emphasised the essential imprecision of arguments for, or against, 'minis', it is possible to categorise the key ones in a useful framework relating to overall 'responsiveness'.

Reliability

Reliability is still a major determinant of responsiveness. The general principle favouring minis is that a computer system is a complex of electronic and electromechanical modules each of which must function properly for the whole system to be operational – the smaller the system, the longer the MTBF. Clearly this principle can be, and is being, applied to large central systems based on multiple machines. However, it applies most readily and naturally to distributed systems of minicomputers.

It is worth emphasising again the human aspects of unreliability, particularly as they affect non-computer-oriented users. A degree of unreliability that is completely acceptable in a batch-processing environment can be devastating for an on-line, interactive system. Many applications require a continuous, 24 hours/day, 7 days/week, service, but, even when a delay in access can be tolerated technically (in terms of required throughput), it can be intolerable to the remote interactive users who are already isolated from the central system and become completely so when it is not available. In particular, they are faced with psychologically disturbing uncertainty in deciding whether to attempt to access the system again and face further rejection, or wait longer even though it may have become available. It is a natural human reaction to attempt to avoid such a situation, and it can be avoided by ceasing to use the system. Since many potential users have no particular enthusiasm for computers, often quite the contrary, any such demotivating influence is a major problem.

Clearly a central service can take steps and install mechanisms to alleviate such problems, for instance, by making the communications network have sufficient capability to continue to send messages to users and accept enquiries even if the main service is down. However, again there is a strong element of trust involved – is the necessary action being taken? – who has priority in getting re-started? – and so on. Action on a local system is visible and questions may be asked, service engineers harangued, etc. The social stability of a distributed system is essentially greater than that of a centralized one under conditions of uncertainty.

Low overhead of interaction

Input/output is generally an expensive overhead on medium-sized mainframes with a

single processor and large word-size; the workload of interrupt-servicing and message-buffering ties up a powerful processor and expensive high speed memory. Interactive data entry, which is of major importance in data management systems with non-specialist users, is an expensive overhead on machines designed for batch number-crunching. Where the workload is predominantly data acquisition then the minicomputer is far better matched to the job. Again, a central service can itself use a minicomputer to 'front-end' its main machine, but this needs to be justified in terms of some other pressing need for centralization – otherwise the simplest, most robust and most appropriate location of such a 'front-end' is with the local users (who then may not need the 'back-end' at all – there are still many tails wagging an awful lot of dogs!).

Short response times

Short response times are an essential requirement for effective interaction. Reliability may be seen as one major contributor to response times but, assuming the system is working, the interactive response-time is a function of load and scheduling system. For a general purpose central system the load can fluctuate wildly and the variable job-mix can make effective scheduling complex to the point of impossibility. Miller (036) notes that psychological 'step-down discontinuities' occur as response-time grows to exceed basic human parameters of short-term storage, attention span, etc, and highlights two seconds as a maximum for conversational access. Doherty (008) notes that studies of the large user community at his laboratory over a period of years show that one second of degradation in system response time leads to a further degradation of one second in user response time: both taken together lead to an overall degradation in individual user productivity. The optimum response time for a skilled user is zero (clearly such a time can always be lengthened if desired – the converse is not true!). A minicomputer with its dense order code (with loss of address scope) in a small local timesharing system can maintain a small number of interactive jobs in memory at reasonable cost and operate on a round-robin time-quantum of 100 milliseconds giving mean and maximum response times of a fraction of a second. Best of all, a minicomputer dedicated to a single user needs no scheduling algorithm and can offer a consistent response time that is subjectively zero! Again, careful tuning of an operating system on a large configuration to the job mix of its local community (if such exists as a stable entity) can achieve very good results, but the effort required to cope with such side-effects of centralizing needs justification and is usually just not available – all but a few installations use standard manufacturers' operating systems that attempt to be all things to all men and result in an inefficient compromise.

Consistent response times

Consistent response times are just as important as short ones. A zero response time is clearly consistent, but a two-second average usually hides a very long-tailed distribution. We have highlighted previously (018) the devastating effects on naive users, in particular of variability in response times. Hesitations in human conversation are key indicators of the significance of the following information (025,031) and time delays between cause and effect are used naturally and subconsciously by people in modelling the behaviour of an interactive system. In a timesharing environment we fool someone into thinking that they are the only user of a system and, most of the time, the system hesitations are causal variables dependent on the transactions they are undertaking. However, on a significant minority of occasions time delays are nothing whatsoever to do with the user suffering them but result from other activities of which he is unaware. Creating a situation where information being fed to the user is usually meaningful but significantly often false is psychologically disruptive. Response time consistency can be improved greatly on a small local system where the workload is known and the causes

of variations are apparent. Again, response time is most consistent of all on single-user systems.

Decentralization of data

Decentralization of data is itself attractive; failures leading to data loss or corruption tend to be more catastrophic with current computer systems than with conventional information management by files and clerks. The centralization of information in a set of files for the sake of efficiency and high workload also maximizes disruption when access to those files is lost. Distribution of workload and files of a large system between a number of independent modules increases the overall system robustness.

This I see as a key problem area for the next decade. Universal access to an overall data base is highly attractive and will be a key feature of most business applications in the future. However, the simplest way to achieve it currently is through a centralized filing system, even if processing power is itself distributed. Hardware/software unreliability may soon become insignificant factors and the basic security of such central files may be technically acceptable. However, the potential for human disruption by labour disputes, acts of sabotage, embezzlement, etc will remain great. Too much power is being concentrated in too few hands. The need to develop distributed data base systems with the same facilities as centralized ones will be forced upon us, not so much by technical factors, but by such organizational and human ones once again.

Duplication

Duplication of capabilities (rather than function) allows a total (distributed) system to operate at full workload under normal conditions but shed load in a controlled manner when failures occur, e g, files and work can be moved from one machine to another which has less important functions. Minicomputer-based systems are now at a stage of weight and compactness where complete physical systems can be moved readily from one location to another to replace defective equipment or cope with increased local demand.

Ease of system operation and management

Ease of system operation and management is increased by the use of minicomputer systems designed for industrial use and not requiring special environments (caveat emptor — this applies to by no means all mini configurations – ask the service engineers not the salesmen!). The complex high speed peripherals on large systems require specialized operator care and attention. By distributing the load to a number of far simpler, low speed local terminals the dependence on such staff is eliminated. Operations such as file back-up may also be designed around the local context allowing security to be ensured and seen to be ensured, by those most concerned about it.

System design

The most important feature has been left to last. The prime cause of problems in most computer system development is the attempt to take on too much and to do everything in one major development. There is a crucial limit in system development at the point where the system as a whole is no longer comprehensible to one individual. 'One task, one person' is a recipe for success (where a 'task' is the design of a relatively independent part of a system). Dividing requirements up into small projects with limited objectives on local machines, and then expanding with more projects and more configurations, leads to rapid, robust progress. Systems tend to freeze with time and successful use, and it is far easier to set up a new system than to increase the capabilities

of an old one. Again, all the benefits previously mentioned of having the relevant staff, this time the system designer and implementor, tied down to a local configuration also apply.

Clearly, these are all *potential* advantages of the use of small, distributed, functionally complete systems. Any of them may be squandered by poor system design, and many of them may be achieved on large central configurations although generally with far greater effort since they do not naturally apply. There are also substantial advantages in large, central systems all stemming from the economies of size possible through sharing resources. However, these are becoming far less significant in view of the declining costs of powerful small computing systems compared with the costs of communicating with a remote central system. It is also clear that new problems of duplication of effort, undue diversification, incompatibility, etc, can arise through the uncontrolled proliferation of local systems. Such new problems must be clearly defined and tackled if full advantage is to be taken of the potential of minicomputers; this, in itself, will happen solely because of market forces rather than technical principles. The argument is not so much that we *should* learn to live with distributed systems, but that we will *have* to, because the advantages they offer far outweigh the disadvantages.

However, as will become clear in the following section, I am not arguing for a future based on diverse, isolated minicomputer systems with no communications and no centralized functions. Unnecessary centralization as a vestigial remnant of the historic development of computer systems (and personnel) is the only target for attack. The need for effective inter-system communication (whilst avoiding unnecessary dependence on it) will always be a key one; we should look to communications protocols for the standards ensuring reasonable consistency across a distributed system. It is not how I maintain my local records, or process them, that matters to you, another member of the network, but in what form I pass them to you when requested. This is not very different from the way in which we currently operate in large organizations: accountants, lawyers, personnel officers, salesmen, etc, each have their own approach to information handling and processing. However, they all (sometimes with great difficulty) manage to converse intelligibly with one another. They also all, to a greater or lesser extent, follow centrally established policies and produce summary reports of their own activities as general management information. What the local mini-based system offers is a natural enhancement of these already established and effectively operating structures that does not entail undesired change in the way they are currently organized.

THE NEXT DECADE

I have noted previously (017) the dangers of attempting to project computer technology very far into the future. We are in an era of rapid technological and social change and the variety of answers possible from a mixture of predictive and normative forecasting is very great. In particular, techniques of man-computer communication are advancing very rapidly and current approaches will appear very antiquated in ten years' time. In these circumstances it is best to place particular projections of minicomputer-based system technologies within a framework of other broader studies of the role of computers in business organizations in the next decade, and to reference specific possibilities to this overall outline.

The key commentator on the assimilation of computer technology into commercial organizations in recent years has been Richard Nolan who has published a range of studies of the impact so far of computer systems on a variety of organizations, their structures and staff (038,039,040,041,042,024). Richard L Nolan (038) has hypothesised that the

assimilation of computer technology is following a series of distinct 'stages': *initiation, contagion, control* and *integration*. He has recently (042) speculated as to the nature of a fifth stage, discussing hypotheses:

1 "Stage V will be the first stage of a series of four that will be similar to Stages I-IV where organizations will learn to assimilate the most recent advances in information technology which include data base, micro/minicomputers and telecommunications" – i e, the new technologies are so radically different from the old that we have, effectively, a new beginning.

2 "Stage V will be the organization learning about assimilating word processing technology" – i e, the specific technology of word processing and its derivatives will itself lead to major changes in business operation.

3 "Stage V will be the organization learning about assimilating micro/minicomputer technology" – i e, the specific technologies of micro and mini computers will themselves lead to major changes in business operation.

4 "Stage V will be acceptance of the reorganization of data processing to appropriately align data resource management functions" – i e, the specific technology of data base systems will itself lead to major changes in business operation.

5 "Stage V will affect different companies in different ways depending on their industry, managements, and the impact of the relevant information technology" – this Nolan takes as a contingency hypothesis (i e, that we cannot make any further generalizations about the effects of technological change), but it may also be read more constructively as saying that the new technologies offer greater flexibility and customization than the old – i e, the style of operation of the organization can be retained even though the new technologies are being assimilated.

All of these hypotheses are consistent with, and substantiated by, the arguments of this paper. The technologies of word processing, the data base, minicomputers and communications, make possible distributed networks of local, but interconnected, processors that can be moulded to, and integrated into, the corporate structure of any organization, large or small. This gives an opportunity to avoid the distortions of structure introduced by the previous generation of centralized, batch-processing systems, and tailor the information processing and distribution network of the company to its actual, or desired, structure. Thus the company structure will drive the computer system structure and not vice versa. This form of close integration of computer technology and human organization is itself a radical change and will go through the four stages Nolan has already identified.

There is one major factor that is, to some extent, implicit in Nolan's argument, but which really merits a separate hypothesis:

6 Stage V will be the establishing of close man-computer relationships in which the peculiar talents of each partner will be enhanced by the other and their individual weaknesses compensated by the other.

Such a dream of man-computer symbiosis has been with us since the early days of computing, but only now is it becoming both a technological and an applications reality. In the remainder of this section I will first detail the current state of widely available minicomputer systems, then outline the currently available technologies that will lead to the next generation of commercial systems, and finally project this towards the end of the decade illustrating how it corresponds to Nolan's hypotheses.

Minicomputer-based systems now

Figure 1 is a diagram of a typical minicomputer-based business system in current use (I am basing this section on hardware and software available now from major manufacturers, fully supported, etc). It is best thought of as an electronic filing system with basic clerical and secretarial facilities. The key features of the system are as follows:

1 *Processor*

mini/micro-processor with 16-bit instruction set in firmware but not dynamically accessible

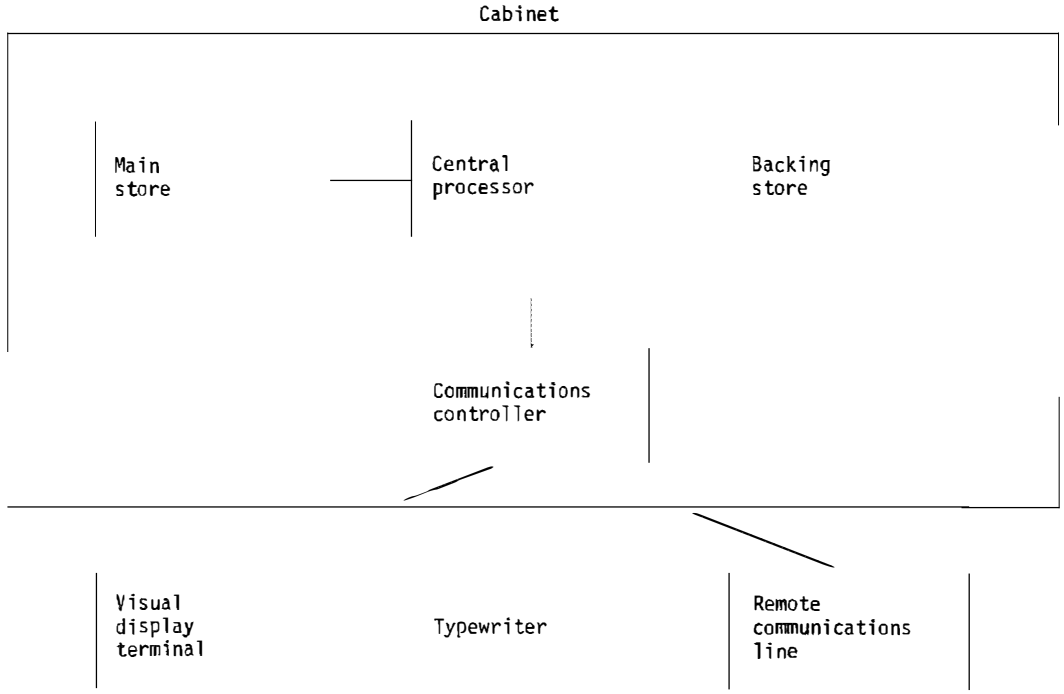

Figure 1a: Minicomputer hardware configuration

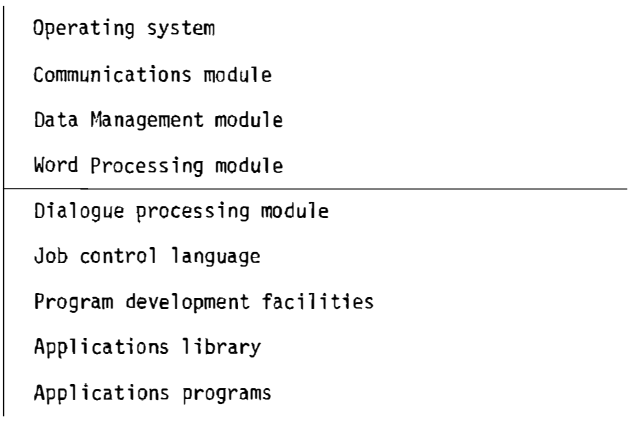

Figure 1b: Minicomputer software configuration

Figure 1: A typical minicomputer-based business system

- multiple index registers giving access to 65K 8-bit bytes of main store through variety of address modes
- wide range of operations on 8-bit character, 16-bit integer and 32-bit real data
- 16-bit add time few microseconds
- current end-user cost: about £1000 in rack with power supplies for processor and basic peripherals.

2 *Main Store*

- semiconductor RAM with access time of about 0.5 microseconds
- no relocation, paging or other memory management system
- current end-user cost: about £2500 for 65K 8-bit bytes.

3 *Back-up Store*

- dual floppy disk unit for program and data filing
- capacity 250K bytes per unit on industry-compatible, replaceable, interchangeable medium
- mean access time 300 milliseconds
- current end-user cost: about £2500 with interface and controller
- media cost: about £4 per disk.

4 *Communications Controller* for interfacing terminals and links to other machines

- multiple-line, variable-speed, asynchronous serial line unit
- current end-user cost: about £500 for 4 lines.

5 *Cabinet* for housing this equipment

- normal office furniture in style with other office equipment
- low-noise ventilation system and no special environmental requirements
- current end-user cost: about £600.

6 *Printer*

- quiet, high-quality typewriter for document output
- 50 characters/sec
- range of fonts on interchangeable wheels of some 95 characters
- variable spacing and simple graphics

current end-user cost: about £2000.

7 *Visual Display Terminal* for system control, data entry, rapid dynamic retrieval and operator training

1000 characters/sec

24 lines of 80 characters

upper/lower case

detachable QWERTY keyboard

current end-user cost: about £750.

8 *Operating System*

device-independent buffered input/output system

multi-level file structure on back-up store

timesharing among multiple tasks on simple round-robin basis

resident system less than 8K bytes (no overlaying since floppy disks assumed)

current end-user cost with basic system utilities, assembler, linker, etc: about £1000.

9 *Communications Module*

network communications protocol system enabling tasks on one system to access files and communicate with tasks on another

current end-user cost: about £750.

10 *Data Management Module*

elementary data base facilities in limited hierarchical or network form enabling tasks to access and manipulate structured data in files

current end-user cost: about £1000.

11 *Word Processing Module*

controlled text output module enabling documents to be formatted

current end-user cost: about £750.

12 *Dialogue Processing Module*

controlling and analysing interactive dialogue and providing terminal user with range of standardized facilities, e g, 'question-in-depth' or 'help' facility.

13 *Job Control Language*

- allowing users to construct nested command files of system commands
- usually regarded as part of 'operating system' but separated here because it is a key requirement in system building and customization.

14 *Program Development Facilities*

- high-level languages and associated linkage, test, and listing and documenting aids
- extended FORTRAN IV with good library as compiled language, and extended BASIC with string-handling and integer arithmetic as interactive language
- modules in both language systems linkable together and with modules written in macro assembler
- run-time overheads typically about 8K bytes
- current end-user cost: about £600 per language system.

15 *Applications Library*

- application-specific modules that are used in a range of applications
- usually written by an OEM supplier partially to customize a system, minimizing the amount of code that has to be generated for specific installations and maximizing standardization of his systems.

16 *Applications Programs*

- the ultimate end-user interface.

Some of the prices above are very approximate since individual manufacturers vary in their 'bundling' strategies, and software costs range typically from zero to twice those given. However, the above summary gives a realistic picture of current, readily available minicomputer technology that is in regular, effective use. In the next section I will use the same headings to discuss the current weaknesses of minicomputer systems, outline the current 'state of the art' variations available (but not yet widely in use or fully supported – these will form the standard systems three years hence), and project past these to the end of the decade.

The next generation of minicomputer-based systems

The type of system outlined above has proved to be cost-effective in a range of business applications. To one accustomed to large mainframes there is usually a sense of shock at the sheer data processing load that such small systems can carry. The main weaknesses of such systems currently stem paradoxically from their strengths; because they are dedicated to specific functions they tend to be somewhat special purpose in concept, designed in isolation rather than to form part of an integrated system. Even the degree of modularity I have indicated in items (8) through (14) above is rare as yet in actual applications although it is now available. Some of the points that follow are concerned

with sheer technological improvement of storage capacity, speeds, etc, but many are also concerned with system organization to allow greater modularity and the use of systems as parts of a larger organization.

Processor

The speed of mini/micro processors is surprisingly adequate for many business applications even when running heavily interpreted programs. However, increased speed can be traded for increased ease of use and will in any event improve with time: there are micros available with speeds 10 times those in widest use and within a few years we shall see 10 instructions/microsecond as standard.

Dynamic microprogramming is also currently available for some minis and micros and will be standard within a few years. This will allow instruction sets to be tailored to language requirements and give a further 10-to-1 increase in speed.

At the same time address range and flexibility of data types will increase so that stores of tens of megabits capacity will be addressable in variable size units from 1 to 250 or so bits; the Burroughs B1700 (051) and Minic-S designs (021) are indicators of these directions.

Main store

The 65K byte addressing range of the majority of micros and minis is again a surprisingly insignificant limitation in a wide range of applications. However, it does place boundaries on future expansion of capabilities that are becoming increasingly ridiculous in view of the rapidly decreasing costs of semiconductor memories. Thus larger stores requiring increasing addressing range in the processor are one obvious extension and costs will continue to fall as 16K and 64K memory chips replace the 4K chips currently in use: a 1M byte memory should cost less than £10 000 in a few years and £1000 in a decade.

Sheer speed in such memories is far less important than overall organization. More powerful processor instructions mean more work per instruction fetched. Data fetches will come to dominate speeds rather than instruction fetches as at present and the use of large cache memories for data pages may become significant in data base applications where keyword tables have to be searched frequently.

Multi-porting memories and segmenting them to allow multiple processors to access them independently with little contention is already being offered for micros in the 'hobbies' market: it is an attractive way of achieving modularity and functional independence by giving separate tasks to separate processors, probably with specialized instruction sets.

Memory protection is one of the weaknesses of the current generation of widely used minis although full protection and mapping schemes are available on some 'top-end-of-range' minis; however, hardware protection and memory mapping are now available for some micros and will become standard over the next few years.

It is probable also that with these quantitative changes in processor and memory costs, speeds and flexibility will come major qualitative changes in computer architecture: 'capability-based' addressing schemes (010) allow robustness of ongoing activities against developmental changes to be assured in complex real-time systems; experiments with 'Interlisp' machines (028,001) are using the technical features to mould the architecture to a powerful language, as has already happened with the B1700's COBOL system.

It is in this area that the future becomes extremely variable since it depends so much on creative imagination rather than the sheer inertia of technological advance.

Back-up store

There are some very strong constraints that I shall place on allowable 'mini-system' backing stores: that they require no special room environment and the medium is simply changeable by naive operators and can be stored in normal office files. This currently rules out all forms of cartridge disks and leaves floppy disks as the main acceptable form of backing store. So many of the advantages of mini-based systems are lost if they do not satisfy these environmental and operational constraints that I feel they must be regarded as inviolable.

However, the current constraints of data capacity and access speed of floppy disks are rapidly being overcome: double-density, double-sided units are available currently that give 1M byte per disk; mean access times have been reduced from several hundred milliseconds down to 90 by use of voice-coil head movements.

A leading manufacturer of floppy disk units recently (CPI 1977) predicted 3M byte capacity disks in a few years, ultimately reaching 10M bytes, and mean access times of 40 milliseconds.

The access time of floppy disks is less significant than their capacity for future projections since bubble memories (050) are becoming available that may be used as buffer memories for the contents of a floppy disk dile, or even the entire disk. Currently a 1M byte bubble memory would cost some £10 000 but this will be down to £500 in a few years and 10M byte bubble memories will be available for this cost by the end of the decade. Access times of such stores can be less than a millisecond with appropriate organization.

The amalgam of floppy disks and bubble memory as a 'datacell' offers a very attractive combination of robustness and speed. It is routine practice to use micros as controllers of such systems and we can expect to see increasing 'intelligence' built into such datacells. There are already stand-alone floppy disk modules offering multi-porting and directory look-up, this will be extended to include contextual searches of keyword tables and files. Again the limits to prediction here are ones of imagination: the datacell will itself become an important architectural feature of new approaches to computer structure.

Clearly, the protection structure, 'capabilities', etc, already mentioned as far as main memory and processor are concerned, should also apply to data structures in backing store. The Graham and Denning (027) model of protection structures characterizes both active processes (programs in execution) and passive data structures as resources with the active ones having access rights specified to all others, active or passive. The ultimate 'datacell' must be a single-level structure reflecting this uniformity. The MUMPS file structure (037) is a good example of the importance to the ease of system comprehension and programming of such single-level structures. It is probably the most significant feature of the MUMPS language contributing to its popularity in interactive system implementation despite the resultant run-time inefficiency with current architectures.

Communications controller

Serial-line communications will continue to play a major role in future systems and the

hardware required for their implementation is dropping in cost as it becomes totally integrated. Its flexibility will also be enhanced with software control of line speeds and characteristics, in-built modems for communication over telephone systems and auto-dial facilities.

Auto-dial facilities allowing the computer system to be programmed to call up other systems are still rare in current applications, but will play an increasingly important role in the future. The switched telephone network both within, and outside, an organization is a generally available utility through which systems can communicate on a casual basis. Thus, there is a sense in which data networks do not have to be designed because they currently exist. Any minicomputer with auto-dial can act both as a stand-alone system and as a terminal to another system, either interactively connecting a local user to a remote system, or interchanging data with it purely under program control (e g at night when phone rates are low).

National packet switching networks may also become available in the next decade that offer data transmission between systems at a higher rate and with lower costs than does the current switched network — the time delays of such systems will be acceptable for many applications although a few will still require direct connection. Within a single physical location there are also attractive possibilities in using a high speed, single-cable 'ring-main' system such as the Xerox Research Laboratories Ethernet (035). Whilst working on a packet basis such a system is fast enough to simulate direct interaction. One can expect the hardware for such system to become routinely available at low cost. In system terms it has the advantage of not imposing any particular network structure on the organization.

Other methods than conventional serial lines are also becoming available for distribution of information to terminals. The Teletext system used by the BBC and ITV for their Cefax and Oracle services on broadcast television may also be used for in-house systems. One line of 40 characters of text (full ISO 7-bit set with upper/lower case and graphics in black/white or 6 colours) is transmitted as one (encoded) line of a conventional TV picture. A standard TV set with a teletext decoder can receive either the picture or a selected group of 24 text lines making up a teletext 'page'. 16 TV picture lines are available (during the flyback period) on which telextext may be encoded so that some 30 teletext frames may be sent out each second (BBC and ITV currently send out only 4 a second). Thus, on each channel of an in-house TV distribution system may be sent out, say, 150 frames of teletext with a mean access time after request at a receiver of 2.5 seconds. The only distribution required is a single, good quality TV ring-main cable. The terminal is a TV set with in-built decoder that can receive conventional television, regional teletext services encoded on it, or in-house teletext from a computer. Such sets are currently available for £500. The teletext encoder at the computer costs about £14 000 plus the cost of buffer memory for the teletext information (1K byte per teletext page). The main disadvantage of teletext is the limited line/page size of 24 lines of 40 characters. However this is a convenient 'window' on data if used properly (as a node in a tree structure). There is actually room for four such pages on the screen of a professional quality monitor so that we may expect to see low cost sets as standard terminals and more complex, multi-frame receivers, as management information terminals. Sets with in-built bubble memories to allow storage of the complete cycle of frames for instant access will also be available.

Cabinet

The importance of minicomputer systems for business use fitting simply and naturally into the conventional office environment cannot be over-estimated but, fortunately,

presents few problems: heat dissipation of systems using semiconductor memories is a few hundreds of watts rather than the Kilowatts of the previous generation, still enough to need a little air-conditioning for the users' comfort in the summer, but not high compared with other office equipment. Noise output from printers has been dramatically reduced in modern high speed matrix printers and daisy-wheel typewriters; thermal printers are available for situations where silent printing is essential. The aesthetics of the computer housing have become important as suppliers aim for executive offices and the home, and very elegant desks housing complete systems are now available that would grace any stately manor! As noted previously, any advances in, for example, backing store technology, will have to maintain this new robust and domesticated status of the computer – environmental constraints will have to be satisfied by, and within, an elegant piece of office furniture that must be capable of operating anywhere a human clerk or secretary might reasonably do so.

Printer

There have already been tremendous advances in printer technology. The old upper-case only, 10 characters/second teleprinters first gave way to 15 characters/second typewriters and then, with a tremendous simplification of the mechanics, to 55 characters/second 'daisy-wheel' printers that can produce a continuous output of high quality documents on conventional office stationery. The basic simplicity of these systems also allows increased reliability and decreased costs as manufacturers proceed along the 'learning curve'.

As typewriter replacements these new terminals are highly competitive already and form the basis of the majority of word processing systems. However, as automatic document producers they are limited by the 96-character set typically available on a single wheel. Changing wheels for different type fonts and point sizes requires manual intervention that limits throughput and wastes effort. There are already machines with more characters per wheel, or with two wheels in use at a time; in future we may expect to see automatic changing of the type wheels become available.

Automatic paper feed allowing non-continuous stationery to be used with printers is also now available. Paper-handling has always been a major problem for office automation but techniques are improving and we may expect to see automatic dispensers for a range of documents become available at a reasonable cost as printer accessories.

Whilst daisy-wheel printers can produce perfect 'camera-ready' copy for lithographic reproduction, any roller-feed system is limited in the maximum point-size that may be used. Low cost photo-typesetters are now available for a few thousand pounds that offer a wider range of faces and sizes than can a conventional printer, although they have the disadvantage of requiring photographic processing. The dry output laser-based phototypesetters now available at the top end of the market will decline in cost as they come into greater production and, whilst not part of a normal office, will certainly revolutionize in-house printing facilities.

Where low-cost printing is more significant than high quality, the dot-matrix printers now available are very attractive, offering speeds of 50 characters/second at a few hundred pounds. Rotating comb printers using special paper give an even cheaper printing terminal (in terms of capital costs) where only occasional hard-copy output is required. We may expect substantial advances in the low cost printer technologies under the impetus of broadcast services for teletext, and an in-built printer will be part of a standard TV set by the end of the decade.

Graphic and pictorial output from business computer systems is far too infrequent currently, due to past technical limitations. Both daisy-wheel and matrix printers are capable of basic graphics using dots at one-fiftieth-of-an-inch pitch. Increasing use of this facility will also place increasing demands on the technology and we may expect to see far more emphasis on graphics/pictorial output in the future. However, this is clearly, yet again, an area where predictions are difficult because available technology is only now beginning to trigger off nascent user demand.

Visual display terminal

Much of the preceding discussion about communications and about printers has already impinged on visual displays. The key direction for mass-produced standard terminals is that already taken by the teletext services: the use of a raster-scan picture output based on TV broadcast receiver techniques. Driving this from a character store gives a low-cost text terminal; driving it from a 'bit-map' in a semiconductor memory gives both a text and a full graphics capability, typically on a 600-dot row by 500 line matrix (this allows about 50 rows of about 80 characters). Using 3 such bit-map stores, one for each gun, allows an 8-colour text and graphics display (using a 1M-bit semiconductor store), and using more bits per gun allows grey-scales and graded colours to be offered. The decreasing costs of semiconductor memories already make such terminals commercially feasible and terminals currently offered cost: £2000 for 48 lines of 80 characters, 192 by 160 dots, in 8 colours; £4000 for 36 lines of 100 characters, 360 by 720 dots, black and white; £5000 for 25 lines of 80 characters, 250 by 5000 dots, in 8 colours; £10 000 for 500 by 600 dots in virtually continuous colours.

In the Viewdata system offered by the British Post Office, a teletext receiver is used as a computer terminal by adding a serial line to the decoder so that the internal text store may be loaded from a remote computer. TV receivers with this 'Viewdata' facility are available commercially for some £600 and may be used as low-cost computer terminals subject to the limitations of teletext format already noted. They may also be used as TV program and public/in-house teletext receivers, giving a very attractive combination of facilities in a low-cost terminal.

Speech communication with computers is a somewhat strange topic to discuss under a heading of 'visual display terminals', but it is logical to do so because speech is a natural competitor to vision in many applications. High-quality speech output from computers has been available at a reasonable cost (some £5000) for many years. Its use has been limited by the need for a reply channel but, with the increasing use of 'touch-tone' telephone systems, communication with a computer over a telephone using voice output and touch-tone input is an attractive, low-cost means of interaction that can compete with VDUs in many applications (053).

Voice input to the computer is far less advanced but already in commercial use for some applications (003). Sound-classifiers that can discriminate between vocal outputs from known speakers using vocabularies of a few hundred words are now commercially available for about £7000. They are currently used for computer input in situations such as quality control, or language routeing, where the operator has no free hand to operate a keyboard. Error rates with skilled operators are remarkably low and, provided there is visual feedback of the word recognised so that correction can be made, are totally acceptable for most applications.

Low-cost speech synthesizers and classifiers are now becoming available in the 'hobbies' computer market, and a rapid advance in the technology may be expected over the next decade now that applications are under way. It is probable that speech 'recognition',

for example, will become one more gadget to aid domestic TV receiver sales, ousting ultrasonic and infra-red remote control units.

Handwritten input of discrete alphanumerics to computers through 'graph-pad' devices is now becoming available with very low error rates in the hands of semi-skilled operators. As with speech recognition, the capabilities of such devices will improve over the decade although the full recognition of cursive script seems unlikely.

Operating system

There is a growing tendency for mini and micro operating systems to be 'hard-wired' into the processors in read-only memories, and much of what has been set about processors, stores, datacells, etc, clearly impinge on operating systems. The hardware/software distinction is very blurred in this context. For example also, the use of microprocessors as peripheral controllers means that what might previously have been a software process is now handled by a separate hardware processor. This in itself helps to promote the modularity and clean separation of functions which is the key trend in modern operating systems. In this respect the very concept of an 'operating system' as a single substantial chunk of software will tend to disappear, 'device drivers' will be regarded as part of the device, 'file structuring' will be regarded as part of the datacell, and even such apparently essentially central features as the resolution of contention for resources on a priority basis can be handled on a decentralized basis by capability-based systems that hand out 'tickets' for the use of resources.

Communications module

The key requirement for communications between systems is that resources on a remote system become available as readily, and in the same form, as on the local system. Thus much of the preceding discussion about resource availability and control also applies to the communications module. However, this does not necessarily imply rigid uniformity of structure across a network. What is required is a general protocol whereby resources of either data or processing on one system may be made available to another. The internal form of the data or processing does not matter provided it can be converted into the form specified in the protocol.

As I have suggested previously, however, the increasing use of mini-based systems in networks will put increasing stress on the ease and efficiency with which the communications protocol can be used and maintained. Thus it is bound to have ultimate effects on both the software and hardware architectures of the local systems. In considering these effects one notes that the protocol will itself be hierarchical with lower levels concerned primarily with accurate communication, and upper levels involving greater semantic presuppositions. This in itself interacts with the concept of data bases, and the network protocol becomes a means for integrating together a distributed data base (043).

The best model for the systems to come is probably that of the message receiving and transmitting 'actors' in Carl Hewitt's Plasma language (048). Graham and Denning's (027) distinction between active and passive processes disappears in this context, since the protection mechanisms for data items are an integral part of their structure making all items active: one process, or actor, sends another a message and the protocol requires it to be able to take note of different standard responses, such as 'actor unable to receive message'. Given such a protocol, it is remarkably easy to write processes that when they require information are able to obtain it either locally or remotely or through processing, talking to different sources through different routes if

necessary. The developments necessary to realize fully such systems are taking place under a variety of headings currently – operating systems, artificial intelligence, networking, etc – but do seem to be converging on essentially simple but fundamentally powerful techniques.

Clearly, I have been making the assumption all along that communication is activity undertaken when one has something to communicate rather than a necessary adjunct to sharing centralized resources. This is not so in most current systems but the rationale for moving in this direction has already been discussed.

Data management module

The management of data has been the theme of much of the preceding discussion – the emphasis in business applications is swinging increasingly away from data processing towards information management – information about finances, resources, staff, customers, sales, deliveries, competitors, etc is generally the key to the mainline business of any organization – much of that information does not need 'processing' in the usual sense of the word, but just routeing to the right place, at the right time, in the right form. Richard Nolan (040) has given an exemplary case history example of the way in which information that is already 'available' on a computer cannot be made use of through lack of an integrated data management system. For many companies the main information of their businesses is still nowhere near a computer system, and only basic financial data is recorded.

Nolan's case history also demonstrates indirectly one of the chief fears of those concerned about the uncontrolled proliferation of minicomputer systems in an organization: you have a local system doing your work very effectively, I have a local system doing mine, but when someone else wishes to use information on both our systems he finds that, even if he can get at it, there are basic incompatibilities, e g, in access keys, that makes it impossible to use both systems together. However, the use of minis does not necessarily introduce this problem and the use of mainframes certainly does not necessarily avoid it!

The commercial use of data base management systems on large machines is itself so recent that few lessons of experience can yet be drawn. Comparable use on (large) minicomputer configurations is even more recent, although some successful systems have been reported (047. The availability of bubble memories will, however, make it possible to implement general, relational data base systems which are both low cost and high speed, and one can expect these to form the heart of future mini-based systems.

Word processing module

The market growth for word processing systems has been phenomenal in recent years and they currently account for the major part of business use of mini and micro computers. However, the production of letters, contracts, etc, is not a mainline business except for a few organizations such as solicitors, and one can expect a swing away from systems that provide only word processing towards *business automation* systems in which word processing is just one function of a system that gives quotations, monitors inventories, generates invoices, etc. Viewed from the opposite point of view, any business system should be capable of producing secretarial-quality documents as its output and hence needs a word processing module.

Current word processing systems have developed from magnetic card, or tape, controlled typewriters and attempt to impose on top of normal use of a typewriter a layer of

special control functions, rather like additional keys on the typewriter. The next generation of word processing systems may be expected to take more advantage of the availability of an interactive computer accessed through a visual display unit and have in-built operator aids accessed through interactive dialogues and incorporating operator training. Document layout and formatting is best done by those who know the function and purpose of the document, so it is attractive to move control of document production further up the organization to those who are unskilled typists but skilled businessmen.

The wealth of knowledge of document production that is available in the printing industry may also be expected to spill over into word processing systems. Automatic hyphenation, for example, follows a complex and perverse set of rules that are, however, already incorporated in computer-based photo-typesetting systems. These will be made available in word processing systems as the cost of the storage needed decreases.

Dialogue

There are already data base access systems that allow users to communicate in truly natural language, i e, not the English defined by the linguists but that used The previous generation of natural language systems required grammatical, unambiguous English such as native speakers rarely produce, but recent systems such as Hendrix's Lifer (030) and Harris's ROBOT (029) can cope with naive users making requests in fragmentary phrases that are meaningful but often not grammatical.

It is reasonable to suppose that such natural language systems, now that substantial advances have been made, will become standard technology during the next decade. Meanwhile, many current systems do not even make their 'unnatural' dialogue as effective as it might be. Facilities to accept arbitrary input formats just as they are already used, or to provide consistently and uniformly means for back-tracking, aborting, or obtaining help, in an arbitrary transaction, should be incorporated in a standard 'dialogue processor'. This will enforce systematic dialogue programming upon the system implementor (018). Much normal office communication already involves stylized messages that can be mimicked on a simple automatic system and this can be very effective provided the human capability of responding to requests for help, breaking off conversations, etc, is also mimicked.

Job control language

Doherty (008) notes that the most widely used high-level language at his Laboratory is Exec II (006), essentially an advanced job-control language. The use of such languages to enable individualized systems to be built up from standard system facilities is now a major factor in the use of many large computer utilities (Doherty also notes that, for example, each user command given on the Stanford Linear Accelerator Center Wylbur system results on average in 25 system commands being executed) and has been made available in minicomputer operating systems such as UNIX (045). Even in the simplest operating system for the PDP-11, for example, the capability of reading nested job-control files is used to allow the operating system to be customized by unskilled users through a simple dialogue (RT11). The same facility may then be used by a system or application programmer to allow the level of users below him to customize the software facilities he is providing. In essence an advanced job-control language provides for a hierarchy of 'virtual machines' (026,013,014,015) down to end-user level. One effect of this is to promote applications software transferability between users since customization can take place with little effort at the new installation, Barnett's COSTAR medical, general practitioner record-keeping system (002) has been made very widely

applicable through the use of such techniques.

Ease of customization with no extra effort on the part of the system supplier is essential if software costs are not to dominate and negate the continuous reductions in hardware costs. Standards, non-customizable packages have been proposed as one answer but take away one of the key advantages of the computer, its inherent versatility and flexibility. End-user 'programming' through dialogue driving an advanced job control language is a far more attractive way of achieving standardization without losing flexibility.

Program development facilities

High-level languages for all forms of computer have totally failed to keep pace with the potential of new hardware developments. The well-established languages, such as FORTRAN, COBOL and PL/1, developed for punched-card orientated, batch-processing environments still dominate current program development. The current generation of systems providing for real-time, interactive use of multiple, asynchronous, interacting processes is not yet supported by any standard language, although clearly many experimental languages have been developed. This is a chicken-and-egg situation where commercial language development cannot be expected until there is a generation of users demanding it, whilst user demand itself is limited by the lack of good language facilities on the systems which offer the new hardware capabilities.

One pragmatic, short-term answer is to make available a wide range of languages capable of being used together on the same data structures (probably with the linkage being done through a job control language that can call modules written in other languages). Standard languages such as FORTRAN, COBOL, PL/1 and BASIC, may be used wherever possible, and non-standard ones such as LISP, POP2, or SNOBOL, used when special demands of list, string, etc, processing make it worthwhile. It could well be that this short-term answer turns out best in the long-term too. With hardware that supports software modularity (allowing rapid switches of context, run-time systems, etc), it should be possible to build systems from comparatively small 'library' modules written in arbitrary languages. This capability to combine old and new language technologies may remain necessary and infinitum since there is no sign of a slow-down in technological advance and change that would allow any 'universal' language to remain so for very long.

For minis currently, FORTRAN and BASIC are generally available, and COBOL is becoming so (or at least reasonable dialects of COBOL) even for micros since business applications (and hence COBOL-orientated programmers) are becoming increasingly significant. Weaknesses in the interactive-dialogue processing capabilities of conventional languages have necessitated the use of special languages such as BASYS (020) for this purpose, and we may expect to see suitable facilities become standardized and more widely available.

Applications library

The ability to extend the family of system modules already outlined with further application-specific modules that contain the common code for a range of products is vital to software production. Proper support of such modularity in both operating systems and languages enables software engineering and production to be treated in the same, well-established framework as hardware engineering and production (016). Quality-control techniques can be applied to small standardized functional modules drawn together into an overall system in a way that is impossible with large amorphous total software systems. This is particularly so if the operating system supports separate segments at

run-time so that modules do not have to be merged through linking and remain available for individual testing within their full final environment.

Applications programs

I have already outlined various features of the operating system, job control language, program development facilities, etc, that make the ultimate applications programming easier, more robust, less costly, and so on. This is the level that must be minimized in its developmental requirements whilst maximized in its end-user impact if we are to make full use of the flexibility of computer systems without undermining the benefits of decreasing hardware costs through an increasing software burden. Techniques of 'structured programming' (005) are of little benefit at the applications level if the system environment in which they operate is itself not well-structured. Proper software engineering requires the right workshop, the right tools and the right materials. The preceding sections have indicated how these may be provided with minicomputer-based systems.

CONCLUSIONS

The projections of minicomputer technology in the next decade given in the preceding section are all reasonably conservative. They invoke no dramatically new hardware or software advances but just predict that what are the frontiers now will have become routine, reliable and fully supported over the next ten years. The diagram of the current situation in Figure 1 is unchanged except for increases in power and decreases in cost, but the system is now seen as one module in a network as well as a stand-alone unit in its own right. This triple role of the system is clearly a key requirement itself: that it be able to take advantage of the network; that it be available to be taken advantage of within the network; and yet that it remain a maximally effective system independent of the network. The philosophy involved is summed up in Galt's maxim:

> "I swear that I will never live for the sake of another man, nor ask another man to live for mine" (044)

with 'minicomputer system' replacing 'man'! Clearly, in any structured network, as in human society, there will be modules that are dependent for their operation on other modules, and modules which have no function in their own right but exist only to serve the needs of others. However, such dependencies reduce the robustness of the network against all forms of disruption and it is probable that robustness itself will be the key design requirement in ten years time when problems of speed, storage capacity, cost, etc, will be far less significant than they are today.

Any major errors in my projections will most probably be due to unpredicted conceptual changes in the structure, function, organization, etc, of computer systems. Clearly, for example, a break-through in man-computer communication, such as direct sensory/motor linkage, could radically affect our basic approaches to computer systems with current work in prosthetics such possibilities are certainly not inconceivable. Even a break-through in voice communication would open up so many new opportunities that much of what we struggle with and require today would become meaningless tomorrow. The impact of computers in the home on a new generation of children, whose familiarity with the technology and its possibilities will breed new attitudes later in life, is itself a new major factor, but perhaps not significant in the next decade.

I hope that the dominant role of human considerations has been apparent throughout the discussions in this paper. There are still fears that computer systems will play a negative role not only in society as a whole, but also locally in the business environment by 'de-humanizing' the tasks of the people involved. Our experience (023) has been quite the converse, that unloading of routine clerical tasks onto a local system tailored to its users leaves far more time for the peculiarly human aspects of business operations. The natural 'slaves' are, and will be for many years yet, the computer systems, although, like all slaves, they react best to being treated well and with respect; the Dickensian 'servant' as the model of an ideal office computer system is argued for in (022). These human aspects of computer systems are increasingly significant as our possible achievements with the technology, and its scope of application, themselves increase. As Josef Kates remarked in his keynote address at IFIP in Toronto, 1977:

> "You are the revolutionaries who will irrevocably transform today's society, and who must communicate to others not only the technical but also the *social, economic*, and even the *philosophical* implications of your discoveries".

Optimism that the technologist is in a favourable situation to make humane use of modern technologies is expressed by Davis, Banfield and Sheahan in (007), where Chapter 9 on 'The strength of the complete technologist' picks out, for example, the following attributes:

> "his methodology in building small models for the operation and understanding of limited areas of work and then assembling strategy" — "ability to achieve reconciliation with the realities of everyday systems" — "accustomed to the use of algebra, calculus, or qualitative and descriptive models for a wide variety of situations" — "used to error and uncertainty and methods for handling them".

These remarks, emphasising small size, modularity, robustness and reality-orientation, pick up the themes of much of this paper. To the extent that we have not built upon these strengths in the past, many systems have been failures, and success in the future depends very much on recognising the factors that Davis *et al* so strongly emphasise throughout their book.

Returning finally to the question of 'minis versus mainframes', I have in essence come down on neither side, since the natural evolution of both minis and mainframes may best be predicted as convergent to the distributed network as outlined here. Within the terms of reference I set, any node within the network that requires a special environment, computer-professional staff, etc, is not a 'mini' — it will in fact have many of the characteristics of current mainframes. The key hope for the future, the key change in approach and attitude possible and required, is that, whatever organization we adopt, whatever nodes we put in the network, it will be because of the needs of the task to be done, *not* because of the needs of the technology doing it. That will be the biggest advance in computer systems and their applications since they were first invented!

REFERENCES

001 ASH W, BOBROW R, GRIGNETTI M and HARTLEY A
Intelligent on-line assistant and tutor system
BBN Report no 3 607Bolt Beranek & Newman Inc Cambridge MA (Jan 1977)

002 BARNETT G O, SOUDER D E, BOWIE J and JUSTICE N
MUMPS: a support for medical information systems
Medical Informatics vol 1 no 3 pp 183 -189 (Sept 1976)

003 COX R B and MARTIN T B
Speak and the machines obey
Industrial Research pp 1-6 (Nov 1975)

004 CPI
Man of the month
Computer Products Intl vol 1 no 2 p 16 (Oct 1977)

005 DAHL O-J, DIJKSTRA E W and HOARE C A R
Structured programming
Academic Press London (1972)

006 DANIELS W E and RYNIKER R W
EXEC 2: a computer language for word programming
RC62 92IBM Thomas J Watson Research Center Yorktown Heights NY (Nov 1976)

007 DAVIS D, BANFIELD T and SHEAHAN R
The humane technologist
Oxford Univ Press Oxford UK (1976)

008 DOHERTY M J
Human factors: impact on interactive computing
SHARE Meeting Cambridge UK (Sept 1977)

009 DOLOTTA T A, BERNSTEIN M I, DICKSON R S, FRANCE N A, ROSENBLATT B A, SMITH D M and STEEL T B
Data processing in 1980-1985
John Wiley & Sons New York (1976)

010 FABRY R S
Capability-based addressing
CACM vol 17 no 7 pp 403-412 (July 1974)

011 FACEY P V and GAINES B R
Real-time system design under an emulator embedded in a high-level language
Proc BCS DATAFAIR '73 pp 285 -291 (April 1973)

012 GAINES B R
Computer technology and its utilization today and tomorrow
Proc Conf on *Small computer applications in industry* Nat Eng Lab East Kilbride UK (March 1973)

013 GAINES B R
Analogy categories, virtual machines and structured programming
In *Lecture Notes in Computer Science* G Goos and J Hartmanis (ed) pp 691-699 GI – 5. Jahrestagung 3 4Springer Verlag Berlin

014 GAINES B R
Human factors in virtual machine hierarchies
Proc Colloquium on the *Influence of high-level languages on computer system design* IEE London UK (March 1976)

015 GAINES B R
Interpretive kernels for microcomputer software
Proc Symp on *Microprocessors at work* pp 56-69 Soc Electronic & Radio Technicians Univ of Sussex UK (Sept 1976)

016 GAINES B R
Hardware engineering and software engineering
Euromicro vol 3 no 2 pp 16-21 (April 1977)

017 GAINES B R
Man-computer communication: what next?
Proc 5th *Man-Computer Communication* Conf Univ of Calgary Alberta Canada (May 1977)

018 GAINES B R and FACEY P V
Some experience in interactive system development and application
Proc IEEE vol 63 pp 155-169 (June 1975)

019 GAINES B R and FACEY P V
Programming interactive dialogues
Proc Conf on *Computing and people* Leicester Polytechnic (Dec 1976)

020 GAINES B R and FACEY P V
BASYS: a language for processing interaction
Proc Conf on *Computer systems and technology* (IERE no 36) pp 251-262 Univ of Sussex UK (March 1977)

021 GAINES B R, FACEY P V, WILLIAMSON F K and MAINE J A
Design objectives for a descriptor-organized minicomputer
Proc European Computing Congress EUROCOMP '74 pp 29-45 (May 1974)

022 GAINES B R, FACEY P V and SAMS J B S
An on-line fixed-interest investment analysis and dealing system
Proc European Computing Congress EUROCOMP '74 pp 155-169 (May 1974)

023 GAINES B R, FACEY P V and SAMS J B S
Minicomputers in security dealing
Computer vol 9 no 9 pp 6-15 (Sept 1976)

024 GIBSON C F and NOLAN R L
Managing the four stages of EDP growth
HBR vol 52 no 1 pp 76-88 (Jan 1974)

025 GOLDMAN-EISLER F
Hesitation and information in speech
In *Information theory* C Cherry (ed) pp 162-174 Butterworths London (1961)

026 GOLDBERG R P
Survey of virtual machine research
Computer vol 7 pp 34-45 (June 1974)

027 GRAHAM G S and DENNING P J
Protection principles and practice
Proc AFIPS SJCC pp 417-429
AFIPS Press (1972)

028 GREENBLATT R
The LISP machine
Working Paper no 79 MIT AI Lab Cambridge MA (Nov 1974)

029 HARRIS L R
User-oriented data base query with the robot natural language query system
Intl J Man-Machine Studies vol 9 no 6 (Nov 1977)

030 HENDRIX G G
Lifer: a natural language interface facility
Proc 2nd Berkeley Workshop on *Distributed data management and computer networks* TID-4500-R65 pp 196-201 (May 1977)

031 JAFFE J and FELDSTEIN S
Rhythms of dialogue
Academic Press New York (1970)

032 KENNEDY T C S and FACEY P V
Experience with a minicomputer-based hospital administration system
Intl J Man-Machine Studies vol 5 pp 237-266 (April 1973)

033 LEWIS B and COOK J A
Toward a theory of telling
Intl J Man-Machine Studies pp 129-176 (April 1969)

034 LICKLIDER J C R
Man-computer symbiosis
IRE Trans on Human factors in electronics vol HFE-1 pp 4-11 (March 1960)

035 METCALFE R M and BOGGS D R
Ethernet: distributed packet switching for local computer networks
CACM vol 19 no 7 (July 1976)

036 MILLER R B
Response time in man-computer conversational transactions
AFIPS FJCC vol 33 no 1 pp 267-277
Thompson Book Co Washington DC (1968)

037 MUMPS
MUMPS-11 language reference manual
DEC-11-MMLMA-B-D Digital Equipment Corp Maynard MA (1973)

038 NOLAN R L
Managing the computer resource: a stage hypothesis
CACM vol 16 no 7 pp 399-405 (July 1973)

039 NOLAN R L
Plight of the EDP manager
HBR vol 51 no 3 (May 1973)

040 NOLAN R L
Computer data bases: future is now
HBR vol 51 no 5 pp 98-114 (Sept 1973)

041 NOLAN R L (ed)
Managing the data resource function
West Publ Co St Paul MN (1974)

042 NOLAN R L
Thoughts about the fifth stage
(1977)

043 PASSAFIUME J J and WECKER S
Distributed file access in DECnet
Proc 2nd Berkeley Workshop on *Distributed data management and computer networks*
TID-4500-R65 pp 114-129
(May 1977)

044 RAND A
Atlas Shrugged
New American Library
New York USA (1957)

045 RITCHIE D M and THOMPSON K
The UNIX timesharing system
CACM vol 17 no 7 pp 365-375
(July 1974)

046 RT11
RT11 V3 user manual
Digital Corp
Maynard MA (1977)

047 SALTER J
the anti-mini myth
Computer Weekly 572
(Oct 1977)

048 SMITH B and HEWITT C
A primer
Artificial Intelligence Lab
MIT Cambridge MA (May 1975)

049 STONEMAN P
Technological diffusion and the computer revolution
Cambridge Univ Press Cambridge UK
(1976)

050 TURN R
Computers in the 1980s
Columbia Univ Press New York
(1974)

051 WILNER W T
Design of the B1700
Proc AFIPS FJCC pp 489-497
AFIPS Press (1972)

052 WITHINGTON F G
Five generations of computers
HBR vol 52 pp 99-108
(July 1974)

053 WITTEN I H and MADAMS P H C
The telephone enquiry service: a man-machine system using synthetic speech
Intl J Man-Machine Studies vol 4 pp 449-464 (July 1977)