SOME NOTES ON THE PROVISION OF DATA ACQUISITION COMMUNICATION, PROCESSING AND RETRIEVAL IN THE SMALL EDP ENVIRONMENT

B.R. Gaines
Man-Machine System Laboratory
Dept. of Electrical Engineering Science
University of Essex
Colchester, Essex, U.K.

April 1976

Introduction

Whilst the conventional image of EDP computer is as a 'number-cruncher' providing large-scale computational resources for a high throughput of numerical calculations, there are an increasing number of small EDP applications whose requirements are primarily those of accurate data-acquisition, communication, storage and retrieval, rather than calculation. Even the classical database computations of search, sort and merge, are not dominant in these applications since the prime requirement is simple storage rather than global analysis.

These systems are best thought of as electronic filing systems with basic clerical and secretarial facilities. The data acquisition system provides clerical aids to facilitate data entry by unsophisticated users. The data retrieval system provides secretarial aids to reproduce the information in the form of well-organized displays and documents tailored to the requirements of the recipient. The actual data-processing required is very small compared with the majority of computer applications. The data storage requirements also are not massive.

Such systems may be seen as a natural extension of conventional office equipment and this is being increasingly recognized by the computer manufacturers who are attempting to serve this market with small computer systems not requiring specialist care and staff. These systems are in competition with: (a) small batch-processing EDP installations; (b) bureau or centralized services; (c) access to remote time-sharing systems. I shall assume in this role a need for interactive data-acquisition ruling out both (a) and (b) as the sole system (a small interactive system can clearly act as a source for larger batch-processors and this is considered).

The arguments for and against (c) are more complex. The economies of size make a centralized utility attractive. So does having all the information available, (subject to protection) to all users. Other advantages such as centralized development of standard programs can, however, also be obtained even if the processing itself is not centralized. Disadvantages are: (a) cost of communication; (b) lower data rate than to local system; (c) poorer response time than with smaller system; (d) more variable system loading gives greater uncertainty in response time and speed of processing than with a smaller system; (e) less reliability than with smaller system; (f) greater vulnerability to disruption than with distributed system.

In practice all four approaches have merits in differing situations. A Combination of time-sharing on fairly small local systems with batch processing on a single large central installation should prove more cost-effective and robust. These notes are concerned with the small local systems, the requirements upon them and the technology currently available.

I shall assume that:

(a) the role of the computer is to fulfil clerical and administrative functions in an existent organization, i.e. we are not building a new system around a computer, neither are we loading 'decision-making' onto the computer;

(b) in so far as possible no specially trained staff should be introduced - indeed the opportunity should be taken to put the ultimate user in as direct communication with the database as they require;

(c) costs should be competitive with human performance of similar functions.

The following sections consider some of the human factors in integrating computers into existent organisations and making them usable by non-specialist staff, and the technical implications of these considerations. Developments

in computers, terminals, storage devices and data-communication are then analysed in this context.

Human factors in computer system design

The close integration of a computer system into an existent organisational environment clearly has its pitfalls and problems:

(1) The organisation already exists and operates as an organic system. Communications and data-processing methods have evolved over a long period that take no account of a computer. The current mode of operation is viable and will tend to have a robustness that rejects change.

(2) There is no possibility of closing down existing procedures and starting afresh. The organisation must continue to operate with full effectiveness during the installation and on-site development of the computer system.

(3) No one person is in a position to explain the current operation of the organisation in detail. Advance systems analysis can only be regarded as an approximation to the actual requirements. It is a way of establishing a reasonable initial integration of the system into the existent procedures and avoiding immediate rejection.

These considerations strongly indicate that the new system should be introduced slowly into the existent situation and evolved rapidly, on-site in response to user criticism. There is one image of major help in visualising the process that the computer system is itself in the situation of a new person coming into the organisation and adjusting to its procedures. The new person will first have to settle down and adapt to the existent environment, to become accepted and trusted as a member of the community. Then, if he has new functions to perform, they will also have to evolve within the existing framework as have previous procedures.

This animistic view of a computer system is very useful in systems design. Many problems can be avoided if they are examined within their social context: "If I were asking you to do this, or giving you information in this form, or responding to your requests in this way, how would you react?" "Would I be prepared to do personally what I am programming the computer to do?". These have obvious connotations in programming the interactive dialogue with the computer which we shall consider in a later section. However, regarding the computer as if it were a new person coming into the organisation also has technical implications, e.g.:

(a) Reliability: a level of up-time completely adequate for throughput may be unacceptable in terms of user reaction. A colleague who was away with frequent irregular illness would tend to be bypassed in routine matters even when he happened to be present. His services would not be fully utilized because his colleagues had learnt to do without them. Thus, even when the computational load in a system is small the necessity of it being immediately available when required demands high reliability.

(b) Response Time: reliability may be regarded as the coarsest factor determining response time. However, even when a system is available its responses must be fast enough to avoid user irritation. In particular, its response time pattern should match user conception of reasonable behaviour. It is no use being very fast in one activity if another, that the user regards as basic and simple, is slow;

(c) Rapid Re-programming system re-programming has to be rapidly achievable because users expect unreasonable behaviour to be corrected when reported and alternative procedures to be demonstrated as discussed.

Again, the new colleague who takes over-long to respond to criticism and adapt his ways to those of the office will come to be regarded as a nuisance rather than a help. The plea so often received nowadays, "Sorry this will now take six weeks instead of a few hours - we have a computer" is just not acceptable if a computer system is to be fully and rapidly integrated into an existing organisation.

(d) Natural dialogue: natural language processing is not programmable for computers at present. However, it is surprisingly unnecessary in the computer performance of clerical duties since most organisations have already evolved stylised, semi-formal modes of communication, presumably to ensure uniformity and comprehensibility. It is these existent modes of communication that computer dialogue must match and natural language itself is not normally required. However, again the animistic view makes it clear that it is the computer dialogue that must be tailored to the existent staff, not vice versa. This calls for a flexibility in text processing possessed by few of the conventional EDP languages such as FORTRAN, COBOL or RPG. This has been the main reason why we have found it necessary to introduce the new language, BASYS.

These human factors considerations suggest a far higher degree of individualisation of computer systems than is conventionally expected. This is not necessary to an ultimate level of different programs for different individual users. Rather the individualisation must be task-orientated. The same interactive package may be used in diverse places for users carrying out the same types of function. That is, organisational standardisation may be desirable and imposable, but not constraints imposed solely by the computer system. Thus there is no reason why many packages

to be run on local systems should not be centrally produced and maintained. However, the initial evolution of these packages requires close collaboration with local users.

The next section examines the role of the minicomputer in relationship to both human and technical factors.

Minicomputer-based interactive systems

Computer systems actually form a continuum from the smallest microcomputer to the largest giant machine. However, there has been a distinction in manufacture and marketing between the EDP machines and the laboratory/industrial minicomputers. This is a multivariate distinction in which one class of machines tends to be used for central systems providing a general service with emphasis on data-processing, and the other class tends to be used for local systems dedicated to a specialist service with emphasis on input-output. Historically, the minicomputer is also associated with lower-cost systems and hence also with lack of software, particularly operating systems and high-level languages, since these tend to be expensive both to develop and to run (in terms of configuration requirements).

It is in the provision of facilities for close man-computer collaboration that the minicomputer offers distinct advantages over the small batch-processing EDP machine. Many workers have analysed relative roles of man and computer in decision-making situations and hypothesized some sort of symbiotic relationship. The computer's accurate memory and rapid numeric and logical process would couple with man's self-organizing, goal-directed strategies, semantic memory and perceptual abilities to generate, through partnership, a system more effective than either alone. Such writers have generally tended to think of the interactive computing provided by 'conversational access' to central 'computer utilities' through 'teleprocessing'. However, considerations such as human factors outlines in the previous section led us to believe that in many situations local dedicated minicomputers are a more appropriate basis

for man-computer systems.

There are a number of basic problems in the design, implementation and application of interactive systems where the use of a number of minicomputers rather than one central system can give definite advantages:

(i) Reliability - a computer system is a complex of electronic and electromechanical modules each of which must function for the system to be operational - the smaller the system the longer the mean time between failures. This factor is important in all systems but it is vital to interactive systems where the degree of unreliability that merely causes a decrease in throughput to a batch-processing service results in an unacceptably low level of availability. Many applications require a continuous, 24 hour/day service but, even when a delay in access can be tolerated technically, it can be devastating to the remote interactive user community who are already isolated from the system and become totally so when it is not available. In particular, they are faced with psychologically disturbing uncertainty in deciding whether to try to access the system again and face further rejection, or wait even though it may possibly have become available. It is natural human reaction to attempt to avoid such a conflict situation, and it can be avoided by ceasing to use the computer-based system. Since many potential users have no particular enthusiasm for computers, often quite the contrary, any such de-motivating influence is a major problem.

(ii) Decentralisation - failures themselves tend to be more catastrophic with computer systems than with conventional information management by clerks and files because, for the sake of efficiency and high

workload, as much information as possible is centralised in the computer's files - distribution of the workload and files of a large system between a number of independent modules increases its robustness.

(iii) Duplication - the duplication of capability (rather than function) means that a system can operate at a 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.

(iv) Low Overhead of Interaction - input/output is generally an expensive overhead on medium-size EDP machines with a single processor and large word-size - the workload of interrupt-servicing and buffering ties up a powerful processor and expensive high-speed memory. Interactive data entry, which is of major importance in data management systems, is an expensive overhead on machines designed for batch number-crunching. Where the workload is not predominantly high-speed, high-precision arithmetic, but rather data acquisition, then a minicomputer is far better matched to the job.

(v) Short Response-Time - this follows from (iv) but is separated because of its importance in human terms - a minicomputer with its dense order code (with loss of address scope) in a time-sharing system can maintain a number of small interactive jobs in core at reasonable cost and operate on a round-robin time-quantum of 100 msec giving average response times of a fraction of a second. This makes the system far easier to use both for naive users who are disturbed by long pauses in response and also for skilled users who are frustrated by them.

(vi) Ease of Operation and System Management - most minicomputers and associated peripherals have been designed for industrial use and do not need special environments. The high-speed electromechanical peripherals on larger systems require most operator care and attention. By distributing the load to a number of typewriter (not teleprinter) terminals or visual displays, centralized operations can be minimized (basically only file security backup).

(vii) System Design - the most important feature has been kept 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 analysis at the point where the system as a whole is no longer completely comprehensible to one person. One task, one person, is a recipe for success (where a 'task' is the design of a relatively independent part of a system). Starting with a small project with limited objectives on a small configuration and expanding with more projects, more configurations lead 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 existent one.

These are clearly all potential advantages of the use of small, functionally complete systems. Any of them can be squandered by poor system design and it is possible to point to large systems that have been well-designed to be superior in all respect to far smaller systems of smaller design. There are also substantial advantages in large, centralised systems all stemming from the economies of size possible through sharing re resources. However, where effective man-computer interaction is the prime objective minicomputer-based systems have, potentially, major psychological and economic advantages.

The New Technologies

There have been several developments in recent years that markedly affect the viability of minicomputer-based systems in small EDP applications. I have chosen the PDP11 as an example but similar remarks apply to other machines.

(a) Fall in Cost of Minicomputers An lsi 11 with 16k 16-bit words now costs only £3k but offers full PDP11/40 instruction set at about one third 11/40 speed . There have been comparable falls further up the range.

(b) Increase in Power Hardware floating point arithmetic on the lsi 11 costs only an additional £120 yet offers a speed increase of over 10 to 1 on software. Thus for EDP work such a machine is comparable in power to an 11/40 (without floating point hardware which costs much more).

(c) Decrease in Memory Costs Memory on the lsi 11 costs £400 for 4k x 16-bits.

(d) Fall in Size A 16k lsi 11 with dual floppy discs is a desk-top sized package (less than 50 cm cube).

(e) Low-cost Backing Store In moving head discs store capacity has been increasing for constant cost so that £6,000 now buys some 10 Mbytes of store. However, there has also been a dramatic breakthrough in low-cost store technology with the floppy discs now available. A 2-transport floppy disc unit costs some £2,000 and offers 256 Kbytes of store per transport (the new models offer 500 Kbyte of store per transport). The medium is similar in dimension to a 45 rpm record and costs about £6.00 each. It is very robust and easy to change. The

main disadvantage of the discs is their comparatively slow access time that can average up to half a second.

(f) Low Cost Displays Visual displays with 20 lines of 80 characters, upper and lower case, are now available complete with keyboard for under £700 from several manufacturers.

(g) Powerful Displays Microprocessor based VDU's offering minicomputer processing power, basic graphics and message switching are now available for under £2,000.

(h) High-Quality Typewriters 60 characters/sec. typewriters with high-quality character sets and variable-spacing are now available at about £1,500. These will take ordinary notepaper as well as continuous stationery and allow compositor-level high-quality documents to be produced.

(i) EDP Interactive Software The RSTS system for the PDP 11 offers a powerful interactive language based on BASIC with full commercial EDP facilities. Configuration costs are £40k - £60k for a multi-user timesharing system.

(j) High-Power Minicomputer Software At the low end of the range RT 11 offers a basic monitor for foreground/background working that supports asynchronous activities so that, for example, multi-user interactive language can run under it. It also supports an extended FORTRAN IV with 16-digit double-precision floating arithmetic and string-handling that is adequate for much EDP work.

These new hardware/software/systems technologies mean that the £40k-60k minicomputer-based EDP systems are very much more powerful, with greater storage capacity, etc., than before. However, they also open up the possi-

bility of distributed systems in which a local machine with about 1 Mbyte of backing store is used by a few local users. Such local machines communicate as necessary to form a network.

The next stage up within the same framework would be to add a very much larger backing store to the local machine - say 10 Mbyte for £6k. However, it might well be that the larger store should itself be regarded as a shared utility, a different type of node in the network. This leads one to the concept of the network itself being in the main entity with nodes on it being for communication with local machines, storage of shared databases, or data-processing probably on a centralized batch-processing system.

Links in the communication network need not exist continuously. In many cases they could be set up automatically as needed to minimize costs. In general the network data rate would not be high because most data processing would be local. The system would be extremely robust because individual nodes would be largely independent and interchangeable.

The design of such a system needs to commence at the local level. How much storage and how many terminals are needed for a particular function? There is a definite breakpoint in technology currently at around 1 Mbyte and 2 to 3 terminals and it is worth examining the maximum work that can be obtained from such a system before going to the next stage.