QUAESTIONES INFORMATICAE

Vol. 1 No. 3

March, 1980

Quaestiones Informaticae

An official publication of the Computer Society of South Africa
'n Amptelike tydskrif van die Rekenaarvereniging van Suid-Afrika

Editors: Dr. D. S. Henderson,
Vice Chancellor, Rhodes University, Grahamstown, 6140, South africa.
Prof. M. H. Williams,
Department of Computer Science and Applied Maths,
Rhodes University, Grahamstown, 6140, South Africa.

Editorial Advisory Board

PROFESSOR D. W. BARRON
Department of Mathematics
The University
Southampton S09 5NH
England

PROFESSOR K. GREGGOR
Computer Centre
University of Port Elizabeth
Port Elizabeth 6001
South Africa

PROFESSOR K. MACGREGOR
Department of Computer Science
University of Cape Town
Private Bag
Rondebosch 7700
South Africa

PROFESSOR G. R. JOUBERT
Department of Computer Science
University of Natal
King George V Avenue
Durban 4001
South Africa

MR P.P. ROETS
NRIMS
CSIR
P.O. Box 395
PRETORIA 0001
South Africa

PROFESSOR S. H. VON SOLMS
Department of Computer Science
Rand Afrikaans University
Auckland Park
Johannesburg 2001
South Africa

PROFESSOR G. WIECHERS
Department of Computer Science
University of South Africa
P.O. Box 392
Pretoria 0001
South Africa

MR P. C. PIROW
Graduate School of Business Administration,
University of the Witwatersrand
P.O. Box 31170
Braamfontein 2017
South Africa

Subscriptions

Annual subscriptions are as follows:

	SA	US	UK
Individuals	R2	$3	£1.50
Institutions	R4	$6	£3.00

Quaestiones Informaticae is prepared for publication by SYSTEMS PUBLISHERS (PTY) LTD for the Computer Society of South Africa.

Thirty Years of Information Engines

G. G. Scarrott

ICL Research and Advanced Development Centre,
Stevenage SG1 2DX, England

Abstract

The era of information engineering was initiated some thirty years' ago by the demonstration of the first stored program electronic computer. At that time there was intense innovative excitement among the pioneers which, however, became somewhat jaded in the subsequent decades as computers became a business and the business acquired its doctrinaire echelons of orthodox systems experts.
More recently, the excitement has begun to return as the triumphant progress of LSI technology has reopened the frontiers of systems engineering so that it is now appropriate to review the evolution of computers with the object of gaining an appreciation of where we are now, by what route we arrived, and what can be expected to happen next. The paper includes brief reference to research into new systems concepts, relevant to such a forecast that are now moving into the market place.

1. Introduction

It is generally understood that the era of information engines started about thirty years' ago with the demonstration of the first stored program electronic computer. At that time, there was intense innovative excitement among the pioneers which, however, became somewhat jaded in the subsequent decades as computers became a business and the business acquired its doctrinaire echelons of orthodox high priests.

More recently, the excitement has begun to return as the triumphant progress of large scale integrated device technology has re-opened the frontiers of system engineering. I therefore propose to outline the evolution of computers with the object of gaining an appreciation of where we are now, how we arrived, what can be expected to happen next and some new system concepts relevant to such a forecast that are now moving into the market place.

A clearly discernable feature of the evolutionary process has been a somewhat fitful trend towards regarding computer system design as an engineering discipline in the conventional sense, characterised by an understanding of design objectives, concern with cost effectiveness, and control of implementation techniques by a subtle combination of theory and practice. Thus, the scope of this paper is probably best described as "the evolution of information engines" — perhaps an unfamiliar phrase but it has the advantage that it not only covers "computers" as we have come to understand the term but also includes foreseeable developments which have not yet occurred.

2. Technological evolution

Let us consider the technological evolution of a typical product in the four phases: birth, adolescence, maturity and senility. When the new product first appears the technology for making it is primitive but, if it serves a useful purpose at all, pioneering users exist whose needs for the new product are so pressing that they are willing to adapt their practices to take advantage of it.

Thus, in the beginning most of the discussion is concerned with "how" to make and use the product and there is very little discussion of "what" purpose the product should serve or "what" should be its technical specification to serve such a purpose. In the adolescent stage, the main population of users begins to appear. Many of these do not have such a clearly recognised need and, indeed, some of them may be only following fashion. As a result of this, discussions regarding the new product begin to tackle the more fundamental issues of "what".

Nevertheless, the technology is still immature so that the adolescent stage can be roughly characterised by the fact that "how" and "what" debates occur about equally, often with regrettably little cross-fertilisation between them. When the product is mature, the technology is fully developed and the market is saturated. All concerned know perfectly well that any relevant product specification can be made so that the crucial questions are entirely concerned with framing a specification which will attract enough users to justify the business. Thus, at this stage, "what" is dominant. Finally, when the product becomes senile it goes out of use and its social purpose is met by other products.

Diagram 1 summarises this maturing process as applied to steam engines, aircraft engineering and information engines. It suggests that information engines are still in the adolescent stage.

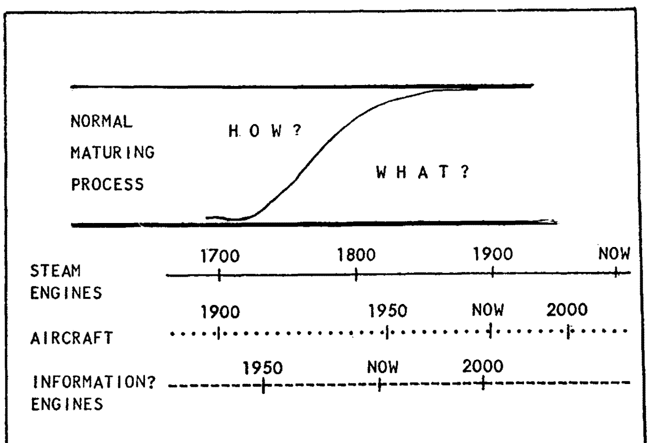

DIAGRAM 1 - TECHNOLOGICAL EVOLUTION

DIAGRAM 1

The first useful electronic information engine was developed during the second World War. Since that time, the physicists and electronic engineers have done a splendid job introducing solid state devices and large scale integrated fabrication techniques which have removed out of sight many of the technological constraints which shaped the early information engines. However, the refinement of technology has not yet been complemented by an understanding of the natural properties of information adequate to guide the deployment of our new found technological mastery so that a first approximation to an understanding of the present situation in information engineering would be to liken it to the situation in the evolution of steam engines in the early nineteenth Century after techniques for casting, forging and machining had provided the "means" but before theoreticians such as Carnot and Rankine had illuminated the "ends" for steam engine design.

3. The present situation in information engineering

Against the foregoing background of general technological evolution, it is useful to summarise the present situation:-

(i) Centralised Data Processing is common.

(ii) Small decentralised data processing systems are proliferating. These contrasting styles of operation are already being widely discussed.

(iii) Interactive use is costly and fragile.

(iv) Complexity is difficult to control. The complexity of an assembly of hardware or software is essentially measured by the quantity of information required to describe it. Complexity is not the same as multiplicity. Thus, for example, if we look at a semiconductor storage chip through a microscope we will see many thousands of components arranged in a highly systematic pattern. Such a storage chip represents high multiplicity but low complexity. On the other hand software comprises almost pure information so that it is characterised by pure complexity. Indeed, human limitations in the handling of complexity now control the range of purposes for which information engines can be effectively used.

(v) Processors originally designed for arithmetic are mainly used for other purposes. We have known this for a long time but done very little about it.

(vi) LSI is used to reduce cost and improve reliability. These are proper objectives but LSI has not yet been used to improve basic system designs which have changed very little over 30 years.

(vii) Device companies are beginning to enter the systems business by offering naked hardware without much software support.

4. Constructional technology

So much for the present situation, what is going to shape the future? It is easy to summarise the technological situation. The essence of the matter is that planar micro-fabrication techniques have rendered the constructional units for processor, fast store, and first level backing store so similar that all three elements can now be assembled in any mixture that we need. There is no longer any over-riding constructional reason for continuing the traditional practice pioneered by Von Neumann of concentrating each type of element, main store, backing store and processor in a separate box and interconnecting the boxes via bottlenecks. Thus, technological advance permits us to design our systems to meet our up to date understanding of the users' requirements but it does not tell us how to do this. Only a deep understanding of the interface between the system and its human users can do this.

5. Essential requirements

It is not so easy to summarise the requirement situation. It is no use trying to analyse the tangled web of computer applications — that way leads only to confusion. Neither is it useful to rake over current systems implementation practice since that has been done many times already. We must start at the beginning. To achieve an understanding of the nature of information and its role in human affairs, adequate to guide the design of an information system, the study should be regarded as a branch of biology rather than mathematics or electronics.

We must recognise that the human race is a species whose original survival and present dominance is based on the chance discovery by our remote ancestors of a new field for biological competition — the creation and operation of social groups dynamically adaptable to the environment by large scale interchange of information between individuals and groups.

We still do this on a grand scale. The hunting and agricultural teams have now become "companies" and the day to day operation of the social co-operation mechanism is called "business". Nevertheless, in our business operations we compulsively adopt organization techniques and associated information handling techniques which served our ancestors for a million years, so that when we introduce a computer system we ignore such inflexible human habits at our peril. However, when computers were first introduced into business the constraints imposed by primitive electronic technology necessitated that to a great extent the user adapt his practices to the computer rather than vice versa. For example, the centralisation of information processing and its collection into artificial batches represented such a forced adaptation. These practices have continued to the present day and have led to many frustrations which have contributed to the somewhat ambivalent image of the computer and its professional attendants as essential but awkward.

We can now recognise that the commercial centre of gravity of information engineering is concerned with "Data Base Management" where the meaning of the phrase should properly be derived from the human realities. The purpose of a "data base" is to assist communication between the members of a co-operating group of people by maintaining a continuously up-to-date information image of the current state of the group and its relevant environment. This purpose can be served only if all the individuals associated with the data base, those who put information in as well as those who access it, feel that the data base is of sufficient value to the group and to each individual to justify its cost. In present practice, a typical data base is not always up to date, it permits its users to ask only stereotyped questions and even to achieve this necessitates a mountain of software which imposes an unacceptable parasitic load on the available computer power and poses a formidable software maintenance problem.

6. Natural properties of human information

It is now possible to see clearly that an effective solution to such problems can be devised only by cultivating an attitude of humility in the design of our information systems. We must first recognise that there is an underlying unity in the tangle of computer applications and that it is derived from their common factor — people. Information structures appear at first sight to be arbitrary and ad hoc — invented on the spur of the moment for each specific purpose. However, such structures are heavily influenced by habits which have been evolved over a long period to create and maintain co-operative social groups. One such habit, the use of tree type data structures, obviously derived from the wide use of tree type social organization structures, is well known to the designers of high level languages but has seldom been reflected in computer design.

A consequential and less obvious habit arises from the fact that for every real situation there are many alternative ways of organizing the associated information in relevant tree structures so that as the situation evolves new tree structures unrelated to those already established tend to be created — grow in importance and then fall out of use. Thus, at any time, many alternative organization structures can be said to have meaning to the people whose co-operation gives rise to the information and it is not practicable to represent all such structures by indexes in their data base.

A third information handling habit arises from our instinctive preference for the interactive mode of information transfer. Although the human species can legitimately be regarded as an information handling specialist and we commonly handle large quantities of information, we can best deal with it in small packets with frequent opportunities for checking and clarification. Accordingly, when we access a data base we reserve the right to alter our question in response to the information obtained. Similarly, we find it impossible to create a large program free of errors, so that every program needs to be debugged before it can be used and, indeed, during use. Curiously, some computer professionals tend to take a puritanical view of programming errors. However, the prevalence of errors is simply the human preference for the interactive mode asserting itself and eventually we can expect that a typical information system will be designed from the beginning to be controlled in the interactive mode.

With this view of the social nature of information, we can deduce some more natural properties:

(i) Most information in the world is not numeric.
(ii) A growing proportion of information processing is not arithmetic.
(iii) Logical operations can be used for all purposes including arithmetic processing.

This has long been known as a mathematical proposition but its practical relevance has only recently been rediscovered and, indeed, we have exploited it in our Distributed Array Processor project. These natural properties of information are summarised in Diagram 2.

Diagram 2

Natural properties of information

1. "Information" bonds society.
2. Information handling habits have been shaped by long use of information to operate social groups, e.g. hunting, agriculture, business.
3. Organization structure of information reflects the structure of the human group which it serves.
4. Hence, Organizational structure of information is normally a blend of order and disorder.
5. People communicate information in small packets with opportunity for checking (interactive mode).
6. Some information processes naturally occur in batches, e.g. payroll.
7. Most information in the world is not numeric.
8. Growing proportion of information processing is not arithmetic.
9. Logical operations can be used for all purposes including arithmetic processing.

7. A technological forecast for information engineering

If we extrapolate from the present situation taking into account the analysis of requirements and available technology which I have outlined, a forecast for the next decade of information engineering is almost obvious. The prime objectives for system design will be data management and keeping complexity under control. The essential technique for ensuring that complexity is manageable is to arrange that the quantity of information handled by the designers or users at one time is within human capability. Thus, the design of both hardware and software must be modular. The modules must not be too large and they must be separated by clean and tidy interfaces such that errors in one module cannot sabotage others. Above all, it will be recognised that universal geniuses do not exist so that the complexity problem cannot be brought under control by seeking more competent people or appointing a new project manager.

A typical system will be designed to be used interactively by people who are primarily interested in carrying on their own business and have no interest in the technicalities of data processing. Batch processing will be confined to operations which naturally occur in batches, e.g. payroll.

At the present time attempts to assess the power of a computer system are somewhat confused by empirical and highly artificial units such as the Post Office Work unit or MIPS. We shall come to recognise that the natural measure of the power of a system is simply the number of people in the organised social group which the system can serve. Systems will be available which will be cost effective serving only a few tens of direct users and they will be purchasable at a cost which will not require user board approval. Such small systems will perform functions which can be clearly understood by the user who will be able to interconnect and redeploy his information engines to meet his constantly evolving needs.

Intrinsically symmetrical communication techniques will be adopted to make this possible. The proliferation of such small systems will take away some of the load of a typical centralised data processing system so that the data processing manager will change his role. Instead of taking direct responsibility for all information processing, he will ensure that the small systems distributed over the users' organisation are compatible with one another and can be used to extract corporate information in addition to serving their primary role as departmental information systems.

8. Some ICL research projects

For several years the work of RADC has been selected to prepare for the situation which I have described. We have put most of our efforts into four projects: Variable Computer Systems (VCS), Content Addressed File Store (CAFS), Distributed Array Processor (DAP), and Interactive Man/Machine Communication by Speech.

8.1 Variable Computer System

The conceptual origin of the Variable Computer System project was the recognition that the natural way that people access information is not by the use of a fixed reference framework as in the von Neumann machine but by the use of a reference map which can be created and continuously maintained up-to-date by the user to suit his purpose. The objectives of the VCS project were two fold:

1. To demonstrate a working system incorporating low level navigation facilities by means of which a user can create and maintain up-to-date a secure reference map showing the organizational structure of his information and its mapping on physical storage.
2. To take advantage of systems incorporating fast microprogram storage by permitting the target machine to be adaptable on

demand to the high level language in which the source program is written.

We now know that the first objective is similar to the capability systems designed at the Universities of Chicago and Cambridge. The provision of a secure map makes impossible many of the programming errors, the unmonitored accumulation of which, accounts for the severe difficulties commonly encountered in the development and maintenance of complex software.

The VCS System has been demonstrable in the Research Department since 1974 and recent work has shown how it could be implemented on standard Company hardware (the MICOS 1 Processor in the 2903).

We have compared the performance of the VCS/MICOS 1 System obeying COBOL programs with the performance of 1900/MICOS 1 (2903) on the same COBOL programs and find that VCS offers:

(i) a reduction in COBOL compiler size in the ratio of 6:1;
(ii) a reduction in COBOL object code size of up to 3:1;
(iii) an increase of COBOL execution speed in the ratio up to 1:1.8;
(iv) a more powerful operating system (in the sense of providing more facilities) which is at the same time more flexible (in the sense of adapting rapidly to different modes of use), with about the same storage requirements;
(v) reduction in main store quotas per job as a result of code sharing and automatic adjustment to working set size.

The complete technique of creating, maintaining, using an information map can be summarised as "access by navigation". This navigation technique is of necessity used in all existing computer systems but in most of them the map is specified completely only in the source version of the program and is irreversibly confused at the time when it is most needed, that is, after compilation. The VCS system preserves the information map explicitly at run time so that accidental deviations from authorised paths on the map can be immediately monitored and controlled. The intrinsic security which these mechanisms permit operate within as well as between programs and serves all the software including the system software, at trivial cost.

Evidently, the use of the navigation technique requires that the work involved in creating and maintaining the map be small compared with the work representing the primary purpose of the information system. This was the case for the tasks for which computers were first used. However, of late it has been increasingly recognised that Data Management can be expected to become the most significant use of information systems. A data base, by its very definition, represents in information terms the activities of people whose interactions cannot be totally predictable. It follows that the maintenance of the information map of a data base involves a very great deal of work to ensure its continued integrity and accuracy as a true representation of the network of agreements, promises and achievements which bind a co-operating group of people. This situation has become widely recognised by those who have been trying to devise standardised navigation techniques for Data Management. Indeed, this recognition no doubt accounts for the controversial nature of standardisation proposals such as CODASYL. This problem is a fundamental consequence of an intrinsically unpredictable component of the natural behaviour of human information so that it is unlikely ever to be overcome within the limitations imposed even by an efficient and secure navigation technique such as used in the VCS system.

It is therefore also necessary to provide a complementary technique to retrieve information by search in those circumstances when the work involved in maintaining the map up-to-date is either excessive or in some cases, intrinsically impossible. There is therefore an unanswerable requirement for a cost effective technique for accessing information by searching for it. The CAFS project to which I have already referred was undertaken to explore ways of providing such a facility and using it in combination with the navigation technique.

8.2 Content Addressable File Store

Over the past few years, we have, in the ICL Research Centre, built such a searching device based upon the use of disk files and we have conducted extensive experiments on methods of using such a facility to meet difficult requirements such as for Telephone Directory Enquiries systems, bibliographic information retrieval, and management information systems. The primitive operations carried out by the searching engine are illustrated in Diagram 3. A selection function is formed from a description of the required data unit issued by a terminal message or by a program. The encoded selector is passed to a backing-store controller equipped with scanning hardware which comprises key-matching channels operating simultaneously on a stream of data, and a special processor which evaluates Boolean or threshold expressions using the comparator outputs as arguments. A wide range of common types of selection function can be represented in a very direct manner in the two-level evaluation hardware.

CONTENT ADDRESSED FILE STORE

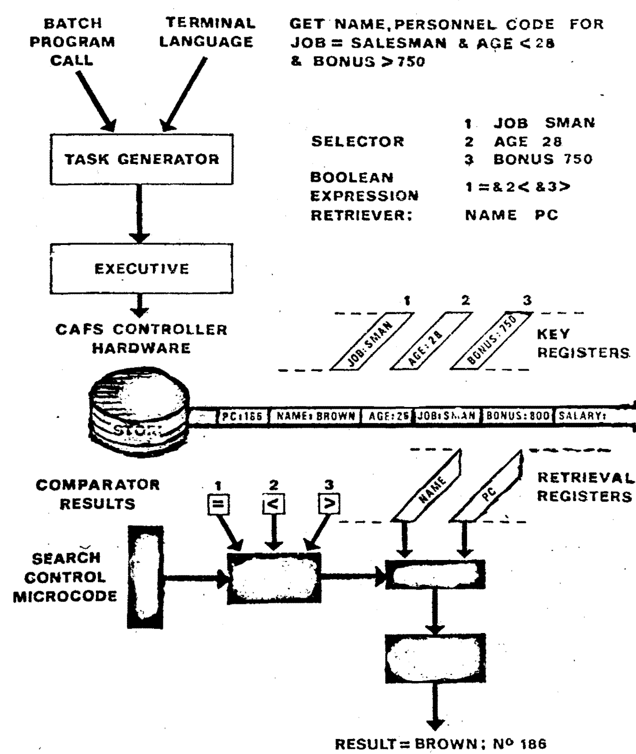

DIAGRAM 3

It is possible for more than one independent search task to be active on the same data stream, the major constraint being the number of key channels available. The latter plug into a standard highway system, and may be thought of as a variable resource analogous to mainframe storage. Each channel is capable of detecting relevant data fields by matching against embedded identifiers, and comparator masking is available to permit part-fields to be isolated. Retrieval of data from "hit" records can be achieved selectively by collecting the contents of designated fields. It is therefore possible to compose virtual "reply records" comprising only that data required by the calling process, arranged in a specified sequence. Alternatively a count of "hits" may be all that is required, in which case, no data as such is recovered. This editing of results is particularly valuable in minimising central resource loading in interactive situations and is generally beneficial in view of the size to which multi-purpose records can grow.

The hardware control facilities are completed by normal physical device controls, including write channel administration. Overall organization is effected by a miniprocessor which is also available to provide a further level of data sieving and composition. The overall balance of search hardware is therefore seen to comprise a filtering mechanism in which the full backing store transfer rate is handled only by very simple, repetitive hardware, with progressively more complex operations being performed on successive abstracts of diminishing volume, culminating in procedures executed in the mainframe. Thus, the sum of the products of data rate and complexity of operation is minimised and, in particular, the mainframe mill and backing store channel load can be reduced by several orders of magnitude compared with conventional serial file processing.

The storage medium can be serial or block-accessed, such as tape or disk. The most generally useful device is the magnetic disk. On a typical high-capacity disk handler many heads are available that require only the addition of some fairly inexpensive electronics to provide a greatly increased read-out rate. Such a high rate would flood all but the most powerful central processor but the progressive abstraction scheme of the special scanning equipment described above renders high speed searching entirely feasible.

Our studies of the use of autonomous file searching devices have shown that it is quite practicable to implement a relational model of a data base. Indeed, we have found it possible to refine the relational model beyond the published work of Codd and his associates.

As I have explained, this project originated in a conscious attempt to identify the facilities required for a Data Base Management system and to provide them by taking advantage of up-to-date technology. We have now reached a stage in which we have studied a variety of specific manifestations of the generalised data base problem and we have not yet discovered any reasons for changing our view of the intrinsic nature of the data base management task.

8.3 Distributed Array Processor

Our third project is the Distributed Array Processor which was initiated about five years ago in direct response to the recognised requirements of weather-forecasting and meteorological research. Since that time, the scope of the work has been broadened to include a very much larger range of problems, e.g. in plasma physics and associative information retrieval.

The great majority of present computer systems can be regarded legitimately as direct descendants of the Von Neumann machine in the fundamental sense that they are characterised by four features:

1. The processor is primarily designed for arithmetic operations with logical operations regarded as a by-product.
2. The store and the processor are separate.
3. The store processor combination can obey only one instruction at a time each requiring not more than two operands.
4. The essential objective in programming such a machine is to represent the overall task by a serial string of such instructions.

These primary features have been somewhat blurred in some powerful machines by tactical measures such as pipe-lined operations on ordered strings of operands but the fundamental principle that strings of individual instructions are obeyed sequentially is still valid. Hence, the connection between store and the processor inevitably imposes a well-defined upper limit to the rate at which the whole assembly can operate. In short, it is a bottleneck.

With the introduction of semiconductor storage and large scale integrated circuits, the original reasons for separating processor from storage are no longer valid. Furthermore, most of the information in the world is not numeric and consequently a growing proportion of computer operations are not arithmetic, so that we should now regard arithmetic processing as a specialised use of more general and fundamental logical operations. Accordingly, to deploy up-to-date technology to meet a broad spectrum of users' requirements, it is now appropriate to revise all four features of established system design practice. In the DAP all these changes have been made.

The conventional semiconductor store is inevitably made in many elements each typically storing a few thousand bits. In the DAP each element has its own very simple processor primarily designed to carry out logical operations on one bit operands. It writes its results into its own storage element and can use as input, information from its own store, its immediate neighbours or elsewhere via row or column highways in a matrix of storage elements. Thus, the DAP offers the following fundamental advantages over current methods.

1. The simple processing elements are easy to design and build, and are flexible in use.
2. The physical distance between each processor can be very short.
3. There can be many such store element/processor element connections operating simultaneously.
4. Real problems are commonly parallel by nature. The DAP provides a parallel processing capability which can match the structure of the solution to that of the problem.
   The DAP-FORTRAN language gives a concise and straightforward way of expressing parallel operations and has already been used to program several applications on the pilot DAP.
5. Since the processing elements are primarily designed to carry out logical operations, a valuable speed up factor compared with present practice is achieved on all operations, data manipulation as well as arithmetic operations. A typical present computer system tends to spend more of its time on data manipulation than arithmetic so that the DAP offers a substantial performance advantage on a wide range of applications.

All the processing elements obey a common program but each element can be instructed or instruct itself to ignore any command in order to provide sufficient flexibility to enable the apparently rigid matrix to be adapted to the parameters of real problems.

The complete assembly can be regarded as a store which has all the properties of a traditional store but with the extra facilities which have been described. It can therefore store its own instructions in the normal way. Moreover, the DAP store can be incorporated as part of the store of an existing host computer of conventional design which is responsible for putting the problem into the DAP part of its own store and getting the answers out. In this way, it is possible to take advantage of the power of the DAP to tackle difficult processor intensive parts of real problems without requiring complementary development of a new operating system.

The proposal was conceived in 1972 and a pilot 32 x 32 has been working since early 1976. It has now become clear from the use of the pilot that the DAP can be applied to a wide range of information processing tasks and, indeed, that intrinsic serialism in real problems is quite rare. Table 1 shows performance estimates for examples from the indicated application areas. A 64 x 64 DAP has been ordered by the Computer Board for use in Queen Mary College and ICL is now actively selling the DAP as an enhancement to its normal system products.

In the longer term, the DAP can be regarded as a new system component, a store with built in processing capability, which is likely to have far reaching effects on the evolution of systems engineering practice such as for example, the efficient implementation of distributed systems.

ARITHMETIC DOMINATED COMPUTATIONS	
Meteorology	13 x IBM 360/195
Magneto Hydrodynamics	14 x IBM 360/91
Structures	6 x IBM 360/195
Simulations	10 x CDC 7600

DECISION DOMINATED COMPUTATIONS	
Table Look Up	3 x CRAY 1
Pattern Matching	300 x IBM 360/195
Operations Research	1200 x IBM 370/145

TABLE 1: 64 x 64 DAP performance estimates on selected problems.

8.4 Man/machine interaction by speech

Our fourth major activity is concerned with Man/Machine Interaction by Speech. A clearly recognisable human habit in the communication of information is the preference for the "conversational" mode. It can be regarded as a behavioural adaptation to the imperfections of human communications since each individual message can be supplemented on request by repetition or clarification.

To carry on a conversation each participant must be able to reply quickly, before the last speaker has forgotten what he said and why. The CAFS system permits such rapid interaction using a keyboard and video display for man/machine communication. Now that we have a machine which can respond fast enough to be a credible conversational partner the ultimate objective is man/machine communication by speech to permit conversational working in the full sense of the word. This is a most difficult problem. The process of human speech communication by natural language is not fully understood and, certainly could not be reproduced by a machine. Our objective is to develop techniques which will enable practical speech communication with an information system to be effected retaining the major advantages of using speech, its ease of use, and efficiency as a means of information transfer.

It is perhaps useful to consider the requirements for a speech input/output system. It should be based upon the use of an ordinary telephone to avoid expensive terminal equipment and to make potential access to a very large number of users. It should incorporate standard digital technology which, on account of its intrinsically high speed compared with the information rate of speech, can be multiplexed to achieve low cost per channel. The use of digital techniques brings the usual advantages of reliability, repeatability and maintainability to what has traditionally been the province of analogue techniques.

The recognition device should be adaptive so that it compensates for peculiarities of the speaker and the individual telephone. In addition to the vocal adaptation the overall system should be designed to simplify the recognition problem by taking advantage of context. At each stage of successful communication in a conversation, the possible repertoire for the next communication is often known to be restricted and there is every reason to take advantage of this fact. By such means it is possible to match a simple machine to a human user capable of great subtlety without excessively annoying the user. Indeed, it is possible to a limited extent to make the machine detect whether a user is experienced, or casual and untrained and structure the interaction accordingly.

Speech output is an easier task for a machine than speech input and is likely to be of commercial significance sooner. We now have ready for exploitation a speech synthesiser, a powerful technique for speech output that can be implemented either in multi-channel or single-channel form. This development has great potential and will enable computer based information services (interactive and non-interactive) to give direct spoken information to the public. For example in a directory enquiry system, about 10 seconds of operation connect time is spent in relaying the telephone number to the enquirer. The use of a speech output device for doing this job would save an estimated £2 million a year in the UK.

Our research activities in the speech interaction field have inevitably caused us to be more aware of the intrinsic nature of information as a by-product of human life. This has been most valuable and will help us to develop techniques for handling the more complex input/output that will be a system requirement of the future.

9. Summary and conclusions

1. A typical information engine will be conceived as a subsystem to a natural human information system. This needs a little explanation. What do we mean by system and subsystem? In conventional usage we think of a computer as a system and its peripherals as subsystems. We used to think we could make a computer any way we liked but a peripheral device must be constrained to plug-in to the computer. We can now recognise that the computer itself is a subsystem in this sense, whose existence can be justified only if it is consciously designed to serve people whose behaviour is unnegotiable. All this will necessitate a new attitude of constructive humility to be adopted by information engineers.
2. We must exploit the order which users instinctively impose on their information and respect the disorder which arises from the fact that human affairs are not totally predictable. We are already well practiced in exploiting order since the design and use of high level languages is essentially directed to this end. However, we can expect to gain much advantage by providing the means for exploiting order at the lowest practicable level in our information system so that they can offer advantages for much of the system software as well as for the ultimate user.
In present practice, disorder is not respected and is too often

regarded incorrectly as a failure of overall system design.

3. Technological advance permits such a system to be designed but does not guide how to do it. Some relevant techniques have been demonstrated in ICL Research.
4. Profound changes in information system practice are inevitable as a consequence of 1, 2 and 3.
5. However, the timing of such changes is difficult to predict since large scale events are controlled by a commercial stick/slip mechanism. This arises from the fact that all large scale decisions are quite properly made to maximise return on investment. The total situation evolves by the accumulation of understanding of objectives together with ripening technology. When such reasons for changes are less than decisive no changes occur at all since the right business decision is to obtain some more return from existing investment. When eventually the reasons become decisive a band-wagon effect occurs so that the changes occur more quickly than might be expected. In my judgement the "slip" is likely to occur in the next decade.
6. Genuinely modular system design will be increasingly practised and will lead to defacto standard functional specifications for modules.
7. A typical module will comprise a combination of hardware possibly including active storage modules derived from the Distributed Array Processor and software whose overall functional specification will be clearly understood by the user in ordinary human terms.
8. The user will be able to control the deployment of such modules to match his evolving requirements.

References

Variable Computer System

1. DENNIS, J.B. and VAN HORN, E.C. (1966). Programming Semantics for Multiprogrammed Computations, *CACM,* **9** No. 3.
2. ENGLAND, D. (1974). Capability Concept mechanism and Structure in System 250, IRIA Int. Workshop, Protection in Operating Systems Aug. 1974, 63-82.
3. EVANS. O.V.D. and MAY, J. (1975). The VCS System — An Overview of its Operational Aspects, VCS.20 RADC, ICL, July 1975.
4. FABRY, R.S. (1968). Preliminary Description of a Supervisor for a machine oriented round Capabilities, ICR Quart. Rep.18, ICR, U of Chicago Sec. 1B.
5. FABRY, R.S. (1971). List Structured Addressing, PH.D Th., U of Chicago.
6. ILIFFE, J.K. and JODEIT, J.G. (1962). A Dynamic Storage Allocation Scheme, *Comput* **J,5,** 200-209.
7. ILIFFE, J.K. (1968). Basic Machine Principles, MacDonald/ American Elsevier.
8. ILIFFE, J.K. (1969). Basic Machine Language, TR. 1021 ACTP Final Report. RADC, ICL 30. Sept. 1969.
9. NEEDHAM, R.M. (1972). Protection Systems and Protection Implementations, Proc.AFIPS FJCC, **41,** AFIPS Press.
10. NEEDHAM, R.M. and WALKER, R.D.M. (1977). The Cambridge CAP Computer and its Protection System, SOSP6.
11. SALTZER, J.H. and SCHROEDER, M.D. (1975). The Protection of Information in Computer Systems, *Proc. IEEE,* **63,** 9.
12. "Variable Computer System", ACTP Final Report, TR.1157 RADC, ICL. Sept. 1974.

Content Addressable File Store

1. CODD, E.F. (1970). A Relational Model of Data for Large Shared Data Banks, *Comm ACM,* **13**,6.
2. COULOURIS, G.F., EVANS, J.M., MITCHELL, R.W. (1972). Towards Content Addressing in Data Bases, *Computer J,* **15,** 2.
3. LIN, C.S., SMITH, D.C.P. and SMITH, J.M. (1976). The Design of a Rotating Associative Memory for Relational Database Applications, *Transactions on Database Systems,* **1,** 1, 53-65.
4. MITCHELL, R.W. (1976). Content Addressable File Store, Online Database Technology Conference, April 1976.
5. OZKARAHAN, E.A., SCHUSTER, S.A., SMITH, K.C. (1975). RAP — An Associative Processor for Data Base Management, AFIPS National Computer Conference, **44,** 379-387.
6. SU, S.Y.W. and LIPOVSKI, G.J. (1975). CASSM: A Cellular System for Very Large Data Bases, Proceedings of the ACM International Conference on Very Large Data Bases, Sept. 1975, Framingham, Mass., 456-472.

Distributed Array Processor

1. FLANDERS, P.M., HUNT, D.J., REDDAWAY, S.F., PARKINSON, D. (1977). Efficient High Speed Computing with the Distributed Array Processor, in *High Speed Computer and Algorithm Organisation,* Academic Press, 113-128.
2. PARKINSON D. (1977). An Introduction to Array Processors, Systems International, Nov. 1977.

Man/Machine Interaction by Speech

1. ADDIS, T.R. (1972). Human Behaviour in an Interactive Environment using a Simple Spoken Word Recogniser, *International Journal of Man/Machine Studies,* **4,** 255-284.
2. ADDIS, T.R. (1978). Human Factors in Automatic Speech Recognition, RADC, ICL. Technical Note TN.78/1.
3. UNDERWOOD, M.J., ADDIS, T.R. and BOSTON, D.W. (1972). The Evaluation of Certain Parameters for the Automatic Recognition of Spoken Words, Machine Perception of Patterns and Pictures, Inst. of Physics Conference Series, **13,** 117-125.
4. UNDERWOOD, M.J. and MARTIN, M.J. (1976). A Multi-channel Formant Synthesiser for Computer Speech Response, Proceedings of the Inst. of Acoustics, Autumn Conference 1976 2-19-1.
5. UNDERWOOD, M.J. (1977). Machines that Understand Speech, *The Radio and Electronic Engineer,* **47,** 8/9, 368-376.

Notes for Contributors

The purpose of this Journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles, exploratory articles or articles of general interest to readers of the Journal. The preferred languages of the Journal will be the congress languages of IFIP although papers in other languages will not be precluded.

Manuscripts should be submitted in triplicate to either Dr. D.S. Henderson or Prof. M. H. Williams at

Rhodes University
Grahamstown 6140
South Africa

Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins. The original ribbon copy of the typed manuscript should be submitted. Authors should write concisely.

The first page should include the article title (which should be brief), the author's name, and the affiliation and address. Each paper must be accompanied by a summary of less than 200 words which will be printed immediately below the title at the beginning of the paper, together with an appropriate key word list and a list of relevant Computing Review categories.

Tables and figures

Illustrations and tables should not be included in the text, although the author should indicate the desired location of each in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Illustrations should also be supplied on separate sheets, and each should be clearly identified on the back in pencil with the Author's name and figure number. Original line drawings (not photoprints) should be submitted and should include all relevant details. Drawings, etc., should be submitted and should include all relevant details. Drawings, etc., should be about twice the final size required and lettering must be clear and "open" and sufficiently large to permit the necessary reduction of size in block-making.

Where photographs are submitted, glossy bromide prints are required. If words or numbers are to appear on a photograph, two prints should be sent, the lettering being clearly indicated on one print only. Computer programs or output should be given on clear original printouts and preferably not on lined paper so that they can be reproduced photographically.

Figure legends should be typed on a separate sheet and placed at the end of the manuscript.

Symbols

Mathematical and other symbols may be either handwritten or typewritten. Greek letters and unusual symbols should be identified in the margin. Distinction should be made between capital and lower case letters between the letter O and zero; between the letter l, the number one and prime; between K and kappa.

References

References should be listed at the end of the manuscript in alphabetical order of author's name, and cited in the text by number in square brackets. Journal references should be arranged thus:

1. ASHCROFT, E. and MANNA, Z. (1972). The Translation of 'GOTO' Programs to 'WHILE' Programs, in *Proceedings of IFIP Congress 71,* North-Holland, Amsterdam, 250-255.
2. BÖHM, C. and JACOPINI, G. (1966). Flow Diagrams, Turing Machines and Languages with only Two Formation Rules, *Comm. ACM,* **9**, 366-371.
3. GINSBURG, S. (1966). *Mathematical Theory of Context-free Languages,* McGraw Hill, New York.

Proofs and reprints

Galley proofs will be sent to the author to ensure that the papers have been correctly set up in type and not for the addition of new material or amendment of texts. Excessive alterations may have to be disallowed or the cost charged against the author. Corrected galley proofs, together with the original typescript, must be returned to the editor within three days to minimize the risk of the author's contribution having to be held over to a later issue.

Fifty reprints of each article will be supplied free of charge. Additional copies may be purchased on a reprint order form which will accompany the proofs.

Only original papers will be accepted, and copyright in published papers will be vested in the publisher.

Letters

A section of "Letters to the Editor" (each limited to about 500 words) will provide a forum for discussion of recent problems.

Hierdie notas is ook in Afrikaans verkrygbaar.

Quaestiones Informaticae

Part 2 of the proceedings of the first South African Computer Symposium on Research in Theory, Software, Hardware, organised by The Research Symposium Organising Committee of The Computer Society of South Africa. 4 & 5 September 1979, Pretoria.

Contents/Inhoud