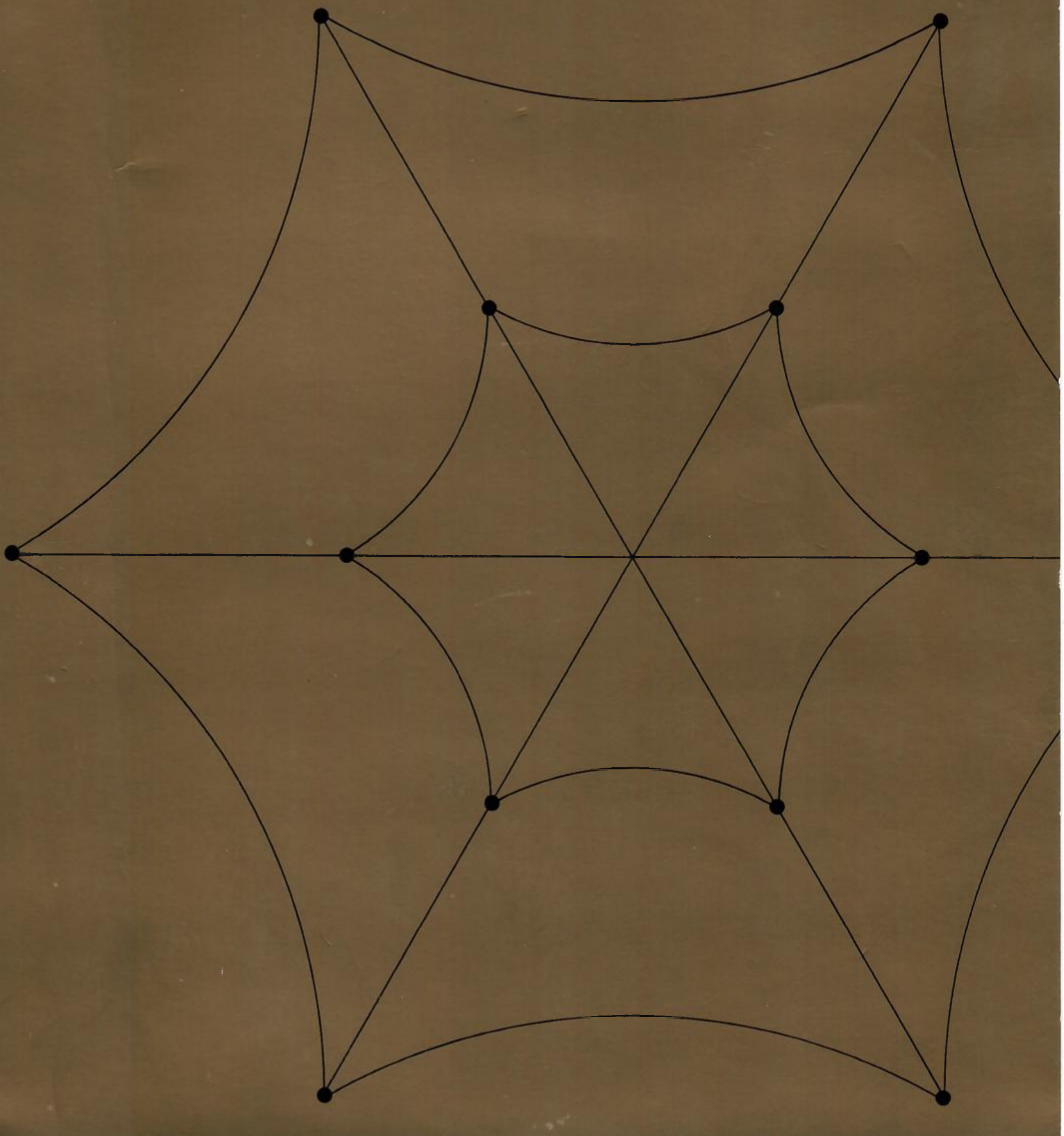
The coordinated programme of research in

DISTRIBUTED COMPUTING SYSTEMS

annual report

September 1982 – September 1983

Science and Engineering Research Council



THE

COORDINATED PROGRAMME

OF

RESEARCH

IN

DISTRIBUTED COMPUTING SYSTEMS

ANNUAL REPORT

Sept 82 - Sept 83

Science and Engineering Research Council

1983

Part I

Table of Contents

1. INTRODUCTION

This is the sixth Annual Report of the SERC's Distributed Computing Systems Programme. Its purpose is to:

1. Outline the objectives of the programme.
2. Define the programme's scope and the mechanisms for implementing it.
3. Describe the projects to an audience outside the programme.
4. Report on the programme's progress to September 83.

In particular, it is intended to stimulate contacts between researchers within the programme and those in industry, research establishments and universities in the UK and abroad who are working in or interested in distributed computing.

Part I of the report describes the background to the programme, the scope of the technical work in the programme, management of the programme and short summaries of milestones in the programme's development. Part II gives descriptions of the activities of each group in the programme, key references and contact addresses for further information. Part III contains financial details of research grants awarded, details of workshops held, an address list of all investigators in the programme and a membership list of the management panel.

2. BACKGROUND

Prior to 1976 projects concerned with distributed computation had been funded for some considerable time by the Council in its conventional response mode. Recognising the importance of this area, the Computing Science Committee appointed a Panel in June of that year to consider what action was necessary to encourage, coordinate or direct research into Distributed Computing. The Panel's recommendations, for a coordinated research programme with additional funding, were warmly welcomed by the Council's Engineering Board, and the programme was initiated in the academic year 1977-8.

Distributed Computing Systems (DCS) research is a Specially Promoted Programme of the SERC. It is monitored and administered by a nominated DCS Panel. The programme will continue for a total of seven years, until 1984, by which time it is expected that some 6,500,000 pounds will have been invested in approving 120 grants to investigators in 26 institutions.

The model of coordination that has evolved in the Distributed Computing Systems Programme has been taken up by newer SERC initatives, for example the Software Technology Initative and the Robotics Programme. The Programme has also had an influence outside the SERC, for example two former chairmen of the DCS Panel were members of the Alvey Committee (1982).

3. OBJECTIVES OF THE PROGRAMME

The primary objectives of the programme are to seek an understanding of the principles of Distributed Computing Systems and to establish the engineering techniques necessary to implement such systems effectively. In particular, this requires an understanding of the implications of parallelism in information processing systems and storage, and devising means for taking advantage of this capability.

The Panel's remit may be described as follows:

1. To achieve results of practical value to UK industry by directing research to a key area for the future.
2. To promote relevant Computing Science research of high quality in a positive manner in academic departments by coordinating the efforts and achievements of individual research teams.
3. To ensure the best use of funds at a time of financial stringency.

Within this context, a Distributed Computing System is considered to be one in which there are a number of autonomous but interacting computers cooperating on a common problem. The essential feature of such a system is that it contains multiple control paths executing different parts of a program and interacting with each other. Such systems might consist of any number of autonomous units, but it is anticipated that the more challenging problems will involve a large number of units. Thus, the spectrum of Distributed Computing Systems includes networks of conventional computers, systems containing sets of microprocessors and novel forms of highly parallel computer architecture with greater integration of processing and storage.

The motivations for and importance of research into distributed computing systems are many and varied. The major ones are:

1. Performance: eventually it will be impossible to increase the speed of a single processor and retain commercial viability. Several processors, cooperating on a single task, will be the only way to greatly enhance performance.
2. Reliability: a fully distributed system should be able to tolerate faults caused by either software or hardware. Hardware faults might be tolerated by having more than one of each critical element. Software faults might be reduced by running different algorithms in parallel and checking the validity of results.
3. Clarity: many problems are naturally parallel. Some problems are inherently simpler if expressed as a set of interconnected and communicating processes. If a problem's solution is expressed in this way it might be easier to provide a proof of correctness for the whole solution by breaking the proof task down into proving the correctness of individual processes and then proving the correctness of their interconnection, as opposed to a 'monolithic' approach.
4. Distribution: in areas such as real time control it is often important that processor power is available where it is required in order to minimise the bandwidth requirements of data paths.
5. Cost: the low cost of microprocessors will allow certain tasks to be performed more economically on sets of microprocessors than on a single main frame processor.

4. RESEARCH TOPICS

The Panel has categorised the research into five major topic areas, representing a progression from fundamental theory to systems engineering implications and novel applications. The areas are:

1. Theory and Languages
2. Resource Management
3. Architecture
4. Operational Attributes
5. Design, Implementation and Application

Each of these topics represents a perspective of Distributed Computing Systems from a different viewpoint. The topics are intended as a guide to characterise the investigations and are not a definitive description of the research problems, nor of their potential solutions. Each area is discussed in greater detail below.

4.1. Theory and Languages

The creation of an adequate theoretical basis for Distributed Computing Systems is regarded as having great importance, in particular to underpin work on language and system design. The problems revolve around establishing adequate methods for representing complex asynchronous systems and the identification, formal definition and mathematical analysis of the properties of such systems. It is also expected that such work could have intrinsic merits by extending the notions of formal systems. Milner and Plotkin (Edinburgh) are investigating the mathematics of non-determinism, and the relation between algebraic calculi for concurrency and proof techniques. Experiments are being conducted with industry with notations and techniques as these appear. Lauer (Newcastle) has developed a formalism (the COSY notation) for describing systems as a set of sequential processes and distinct resources. Systems can be analysed for their adequacy, non-starvation, degree of concurrency and distribution. A computer based support environment for COSY is being built and can be used to simulate the execution of a distributed system of synchronised sequential processes, given a COSY source input.

Research is required to identify methods of representing concurrency within programming languages and to evaluate the effect of asynchronism at different levels. There is a need to investigate the extent to which concurrency needs to be implicit or explicit and to determine the constraints a language should place on concurrency to ensure good programming practice. Bustard and Elder (Belfast) and Welsh (UMIST) have developed the language Pascal Plus and compilers for a number of machines. Abramsky and Bornat (Queen Mary's College) are developing the Pascal-M language and program construction methodology to facilitate the design of practically effective user interfaces. The language is being used by Coulouris (QMC) for the construction of interactive systems.

Present programming languages are highly sequential. Concurrency, where it is achieved, is usually obtained by the use of operating system facilities not included in the programming language. There is also a requirement to consider how existing programming languages need to be modified or extended in order to provide facilities in a distributed environment. A major issue is whether the creation of processes and their allocation to processors should be static or dynamic.

Hoare (Oxford) and Hughes and Powell (UMIST) are investigating the techniques of programming by interconnecting networks of communicating processes to achieve simpler solutions which are more amenable to proof as well as efficient, parallel implementations. More radical approaches to the expression of provable parallel programs are the single assignment language ideas of Wadge (Warwick) and the data flow schemata. Zero assignment or applicative languages are being investigated by Darlington (Imperial), Sleep (East Anglia), Henderson (Oxford) and Turner (Kent). Turner has developed techniques for implementing functional languages (combinator reduction) which ease adaptation to parallel hardware. Turner has also developed a functional language, KRC, which incorporates ideas based on set theory, which greatly increases its programming power. Such functional languages show a surprising potential for both parallel evaluation and amenability to proof techniques.

4.2. Resource Management

Distributed Computing Systems do not necessarily eliminate the need for operating systems. The pressure to over-optimise operating systems code may be reduced but new dimensions are added to the problem of resource management. The most important are distribution of control and allocation of logical and physical resources, the scheduling of resources given the additional constraints of access, delay and bandwidth to remote resources, and the organisation of heterarchical systems in which there is no fixed delegation of control. A distributed domain model for systems with no shared memory which allows processes to migrate from one processor to another as a computation proceeds was developed by Shelness (ex Edinburgh) though regrettably this project has now terminated.

Access to information controlled and organised by other processors is a key problem. Distributed Computing Systems exacerbate many of the current database problems, such as control of access, consistency and naming, as well as providing a more severe environment for access synchronisation.

Two contrasting approaches are emerging. One is the server, a concentrated resource such as a filestore, which is accessed by or 'serves' a community of users. Examples are the file servers at Edinburgh and Cambridge. The second, contrasting approach is the truly distributed resource. For example, Bennett (Keele) has built a distributed filestore which appears as a conventional, local filestore to the user but is implemented using multiple copy techniques. More recent work is concerned with majority consensus algorithms and the problems of network partitioning.

4.3. Architecture

A variety of new architectural organisations using multiple computing elements are being evaluated. The use of a collection of computers or micro-computers to provide facilities similar to those on a conventional main frame computer, by distributing identified operating system and user tasks to individual computer elements, is regarded as an intermediate stage of development Such developments will enable existing operating systems and software concepts to be implemented with a greater degree of parallelism on specially organised hardware.

A variety of different architectures are being considered within the programme. Needham et al. (Cambridge) have developed a wide-band digital communication ring attached to which are a number of processors for specific functions. Aspinall (UMIST) is investigating the properties of a 16 processor shared memory system.

A collection of computing elements organised into a regular array, or some other structure, might also be used to provide a general computing facility. Research is necessary to determine the desirable properties of such elements and the form of their intercommunication and interconnection. The potential efficiency of such systems needs to be studied, together with methods of organising computations to exploit the structure and the types of computation for which they are appropriate. Burton and Sleep (East Anglia) are looking at architectures that will give a near linear speedup for divide and conquer algorithms.

It may also be possible to increase the degree of parallelism within a processing element, for example, by the use of data-flow techniques. Several groups (Darlington, Gurd, Randell, Sleep) are looking at non-von Neumann architectures. Gurd and Watson (Manchester) have built a prototype ring-structured data flow computer system. Preliminary performance results are encouraging - without hardware tuning performance improvement through concurrency almost linear for up to 10 processing elements has been achieved. Darlington (Imperial College) is developing a parallel graph reduction machine, ALICE.

Distributed Computing Systems have increased the emphasis on interfaces and interworking. There is a need to evaluate communication protocols, like X25, and to place the design of intercommunication techniques on a sound theoretical footing. The relevance of telecommunication protocols and principles requires investigation. For the longer term, there is a need for a more general consideration of the requirements for effective interchange of information between computers. Mitrani (Newcastle) and Paker (PCL) have developed modelling techniques to help understand the problems of complex intercommunication.

4.4. Operational Attributes

Distributed Computing changes many aspects of system reliability in that the provision of parallelism poses new problems whilst solving others. General methods of using distributed techniques to enhance reliability need investigation. Areas requiring re-consideration in the new context are error detection, damage assessment, fault treatment and error recovery, particularly with respect to performance, extensibility and flexibility. Theoretical and practical studies to define and measure the performance of Distributed Computing Systems are also required. The major grant in this area is the one awarded to Randell (Newcastle) to study the reliability and integrity of distributed computing systems.

Mitrani (Newcastle) has developed performance models for both Ethernet and Cambridge Ring local area networks. A study of program execution by many processors has also been completed. Hutchison and Shepherd (Strathclyde) are looking at experimental comparisons of Ethernet and Ring performance.

4.5. Design, Implementation and Application

Hardware and software techniques for developing and implementing Distributed Computing Systems are needed. The availability of low-cost computing may facilitate the implementation of more sympathetic interfaces for the human user. The study and evaluation of the human interface is therefore regarded as an important aspect for the effective design of Distributed Computing Systems.

The approach to implementing computer applications may be changed by the advent of Distributed Computing Systems. Comparative studies are required to determine the relative advantages of distributed and centralised techniques in various areas. The use of Distributed Computing Systems may enable the application of computing techniques in new areas. Theoretical and practical studies are needed to assess the potential of new applications, particularly in the areas of instrumentation and office automation. Distributed Computing Systems can change the cost of computing and the way in which it can be delivered to the user. Such changes may be expected to have long-term social and economic consequences which need assessment. Cain (PCL) is investigating networks for instrumentation applications and Sloman and Kramer (Imperial College) are investigating techniques for constructing and managing large distributed real-time systems such as are found in control applications. Two grants at Queen Mary College are aimed specifically at office automation. Coulouris is interested in effective man-machine communication and is trying to identify the hardware and software requirements for highly interactive information processing systems. He is using the 'paperless office' as a vehicle for this research. Page has a grant to develop a high quality display system for this application.

Improved techniques for specifying the requirements, by a language or otherwise, which a given Distributed System is to meet are needed, as well as improved hardware and software techniques for translating this specification into a complete system. Cunningham and Kramer (Imperial College) are developing a software tool suite to support the production of verified software. A prototype specification language is under development and is being explored through case studies.

The modularity inherent in Distributed Computing Systems will provide for improved implementation. However, new problems will arise in system testing, and investigations will, therefore, be required into practical methodologies for testing.

5. MANAGEMENT OF THE PROGRAMME

A coordinated approach to research in Distributed Computing Systems has two main advantages:

1. Reflecting the importance of the subject to the progress of computing and information technology, it helps to ensure a reasonable balance of SERC support across the various areas concerned; and a framework facilitating take-up of research results by industry.
2. The substantial costs of much research in this field, and the limits of funds available, make it essential to provide support in a cost effective way - without impinging on the necessary freedom of investigators in carrying out fundamental research.

The coordination is achieved at two levels. First within the programme itself, there is continuing contact with the investigators, starting with assistance in working out proposals with investigators. This is continuing through the research by information services, including a regular research community mailshot, electronic mail, and a wide variety of collaborative workshops. The coordination team is available throughout for *ad hoc* assistance with technical and organisational difficulties.

Secondly, outside the program, links are fostered with industrial organisations (including government establishments, etc.) likely to make use of research results. This aims at the transfer of technology to the industrial sector through involvement of investigators in industrial problems, the exploitation of research products and concepts, and the furtherance of projects through co-operative research. These aspects become of even greater importance as the programme approaches completion.

There exists, then, a positive programme of research into Distributed Computing Systems, managed by a Panel. The Panel monitors the activity of the programme, and advises the Sub-committee on questions of policy and finance. (Up to 1982, the Panel possessed delegated powers for grant awards; those have now lapsed.) The Panel is advised and supported by the Coordination team, who act as 'roving ambassadors' for the programme, and provide day-to-day support for researchers.

The coordination team includes an Academic Coordinator, responsible principally for liaison with and monitoring of the research projects; and a (part-time) Industrial Coordinator, also charged with expanding the external contact range of the research, with a view to technology transfer. The coordinators are supported by a Technical Secretary, and by various support and development staff maintaining and enhancing the programme infrastructure.

6. THE COORDINATION ACTIVITIES

The precise mechanics of coordination continue to evolve as the programme develops, but the essential features are as set out below.

6.1. Academic Coordination

Within the programme, as its scope is very broad, the degree of coordination appropriate to each research project varies according to its nature. Effort is concentrated on the following aspects:

1. The minimisation of duplication of funding by holding informal discussions with investigators prior to submission of formal applications.
2. The promotion of ongoing liaison between research teams and the Panel, via the coordinator.
3. The ongoing review of the objectives of each project in the light of the results obtained, and the progress of the programme as a whole.
4. The review of progress, dissemination of information and promotion of communication and collaboration between investigators via the regular SIG meetings, Workshops, Colloquia, Mailshots and communications facilities.

6.2. Industrial Coordination

As the programme matures, the emphasis lies increasingly on fostering industrial interest in the programme, with the objectives of promoting collaborative projects and facilitating a high degree of technology transfer. This is best achieved via:

1. Invitation of interested industrial researchers to appropriate technical meetings and conferences run by the programme, and organisation of meetings specifically for industrialists.
2. Development of informal links between specific companies and research groups.
3. Subsequent development of links such as research contracts or collaborative programmes between university and industrial or government organisations.

The industrial coordinator provides information on the research programme, advises on mechanisms available for supporting collaborative research, and helps mediate potential researcher/industry relationships.

6.3. Rutherford Appleton Laboratory Support

The SERC supports academic research by awarding grants to investigators for staff, equipment and travel. Investigators may also apply for practical support from the various SERC laboratories which provide specialised services and facilities such as the FR80 colour microfilm recorder, large mainframe computers and microelectronics design and fabrication facilities.

The DCS Programme has been supported by the Rutherford Appleton Laboratory (RAL) since January 1978. The Academic Coordinator and Technical Secretary are on the staff of RAL and other RAL staff provide software and hardware support for the Programme. Typical ways in which RAL support has been provided include:

1. Support for the Unix operating system used by the majority of DCS investigators.

2. Provision of software to couple the Unix troff text formatting software to SERC's III FR80 microfilm recorder.
3. Assembly and distribution of software to drive the Cambridge Ring.
4. Construction of 6 Cambridge Rings for the DCS equipment pool and procurement from industry of a further 10 6-node ring systems.
5. Procurement, distribution and maintenance of the equipment pool.
6. Support and operation of the Programme's electronic mail facility.

RAL has also placed EMR contracts on behalf of the Programme for Unix X25 communications software and a Pascal Plus compiler. The Pascal Plus compiler has been developed at the Queen's University of Belfast, and versions for several machines are now available. X25 communications software for the Unix operating system has been developed at the University of York. Details of both products are available from the DCS Academic Coordinator.

7. SUMMARY OF PROGRESS

7.1. Sept 78 - Sept 79

By September 1978 the DCS Programme consisted of 24 projects totalling 1.4M pounds. During the year Sept 78-Sept 79 the SERC, acting on the DCS Panel's recommendations, awarded a further 16 grants and 2 Visiting Fellowships worth 906K pounds. The extra funds awarded were an indication of the importance attached to the DCS initiative.

During this year the 16 processor CYBA-M system was successfully commissioned by Prof Aspinall's team despite the move from Swansea to UMIST; Prof Randell's team were paid the compliment of being invited to California to present their course on reliability; York's Modula compiler was distributed world wide and was selected for the KSOS secure operating system project.

7.2. Sept 79 - Sept 80

By September 1979 the Programme had grown to 31 distinct projects representing an investment of 2.3M pounds. SERC awarded a further 13 grants and 6 Visiting Fellowships worth 263K pounds.

In this year a Workshop on applicative languages was held in Newcastle which brought together researchers in applicative languages, reduction and dataflow machines which was to lead to at least four grant applications and research projects which have produced very significant results.

7.3. Sept 80 - Sept 81

Nineteen grants and 3 Visiting Fellowships worth 1.25M pounds were awarded to the Programme. Eleven of the grants awarded were 'second round' grants to existing groups within the Programme.

Professor Grimsdale's POLYPROC system at Sussex became operational with three workstations. Close cooperation was established between the groups working in the field of closely coupled systems.

The Programme was well represented at the Second International Conference on Distributed Computing Systems, held in Paris, 1981.

7.4. Sept 81 - Sept 82

Twelve grants and eight Visiting Fellowships worth 861K pounds were awarded to the Programme. Eight of the grants awarded were to existing groups within the Programme.

A significant technical event during this period was the successful execution by Drs Gurd and Watson's (Manchester) prototype dataflow computer of its first programs. Very encouraging results are being obtained.

Some of the results of the Programme to date in the key areas of novel architectures and systems built on local area networks were presented at a Conference organised by W D Shepherd at the University of Strathclyde, on behalf of the DCS Panel. Informal proceedings of the Conference are available from the Academic Coordinator.

7.5. Sept 82 - Sept 83

Twenty three grants and one Visiting Fellowship worth 1.53M pounds were awarded to the programme. Fifteen of the grants awarded were to existing groups within the Programme.

A Conference for senior technical management in UK industry was held at the NCC in Manchester. Some 120 delegates attended. Proceedings of the conference are available from the Academic Coordinator. This volume contains both the reviews of key technical areas which have emerged in the programme (such as local area networks, non-von Neumann architectures and declarative languages) and also reports from individual projects (MININET at PCL and the Manchester Dataflow Project).

Later in the year a technical conference was held at the University of Sussex, organised by Professor R L Grimsdale, at which some 20 research groups presented results.

8. TERMINATION OF THE PROGRAMME

The DCS Programme will terminate in September 1984. It is appropriate to mention in this report that during the final year of the Programme the emphasis will be on technology transfer from the Programme and bringing the Programme to an orderly conclusion. It is intended to hold a conference in September 1984 to mark the termination of the Programme and to present the research results. Details of the conference will be available from the Coordinators.

Part II

INDEX

Bath

PROF J P FITCH and DR P J WILLIS
THE UNIVERSITY OF BATH

FEASIBILITY STUDY OF A MULTIPROCESSOR ARCHITECTURE
Feb 82 - Jan 83

Background and Nature of Research

This project arose from a need to evaluate tree-represented pictures rapidly. It was soon realised that the architecture which developed was amenable to a wider class of problems exhibiting parallelism and this has come to dominate the research.

Broadly the multiprocessor system is organised as a binary tree except that the network is closed by interconnecting what would be the leaves and the root. For certain numbers of processors, this is possible in a completely regular fashion so that the network looks the same from any node. Each node interfaces to exactly three neighbouring nodes and each node consists of a single processor. The closing of the tree is done so cycles on the structure are as large as possible to ensure that a given processor can spawn sub-tasks with a low probability that these tasks will return to the initiating processor. The nature of the interconnection scheme is described in detail in [1].

Research Goals

The SERC grant awarded to the project was for a one year feasibility study. Modest simulation studies have been performed, followed by the direct implementation of the six processor machine. Further experimentation has centred on programming the actual hardware, thus forcing the investigators to adopt realistic solutions to problems such as message passing, task division and so forth.

For the same reason the investigators wished to use a relatively powerful processor at each node, so that potentially interesting results would not be masked by, for example, the addressing limitations of 8 bit micros. Accordingly, the investigators set out to build a six node machine based on 1/4Mbyte Motorola 68000 computers.

Achievements

A six processor net, with interrupt-driven parallel corrections, has been constructed. A dual floppy disk has been added to permit stand-alone working. disk storage is currently being supplied by an interface to a separate machine.

On the software side, a Tripos operating system implementation is running satisfactorily to give a development environment, and a version of Cambridge LISP developed in Bath for the M68000 is running. A simple task spawning mechanism has been demonstrated.

Work in Hand

Work is progressing along two lines. Currently, most activity centres around a conversion environment with explicitly indicated parallelism, but there is also a long term interest in automatic concurrency analysis within Lisp.

The somewhat ad-hoc disk access and the need to use one of the processors as the development machine are current practical difficulties.

Dr Jed Marti has been a SERC Visiting Fellow for the latter six months of 1982. His intimate knowledge of LISP-like environments and grasp of hardware limitations has made his contribution to the project substantial and the investigators wish to highlight the value of such Fellowships. Dr Marti's LISP concurrency analyser is expected to be used soon as a basis of a concurrent compiler [2].

References

1. A.Bowyer, P.J.Willis and J.R.Woodwark, 'A Multi-Processor Architecture for Solving Spatial Problems', *Computer Journal*, 24, 1981.

2. J.Marti, 'Compilation Techniques for a Control-Flow Concurrent LISP System', *LISP Conference*, Stanford University, 1980.

3. J.D.Marti and J.P.Fitch, 'The Bath Concurrent Lisp Machine', *Proc. Eurocal'83* (to be published in *Springer Lecture Notes* series).

Belfast (Queens University)

DR D W BUSTARD and DR J ELDER
THE QUEEN'S UNIVERSITY OF BELFAST

THE DESIGN, IMPLEMENTATION AND APPLICATION OF LANGUAGES FOR DISTRIBUTED COMPUTING
Oct 79 - July 83

The programming language Pascal Plus [1,2] has emerged from refinement of a number of experimental notations developed for large-scale sequential and concurrent applications in the early 1970s. The language can be used to construct simulation models for any concurrent system and to implement such systems on either single processors, or multiprocessors with shared memory. The language is also useful for general purpose programming as it supports Pascal as a sub-set.

The main purpose of this project, together with a related development contract [3], is to make Pascal Plus widely available as a tool to aid research in all areas of computing. This entails the development of a portable compiler kit and specific high quality compilers for three of the leading main frames in UK universities (DEC, CDC and IBM).

All four systems have recently been aligned with the ISO Pascal Standard and both the portable and CDC systems are available for distribution. The development of the IBM and DEC compilers will be completed shortly as part of a new project.

For further information contact: Dr D W Bustard, Dept of Computer Science.

References

1. J.Welsh and D.W.Bustard, 'Pascal Plus - Another Language for Modular Programming', in *Software - Practice and Experience*, Vol. 9, pp 947-957, 1979.
2. D.W.Bustard and J.Welsh., 'A Tutorial Guide to Pascal Plus', Department of Computer Science, Queen's University, Belfast, 1982.
3. J.Welsh and D.W.Bustard, 'Portable Pascal Plus', in *Proceedings of Pascal Experiences Conference*, Gothenburg, Sweden, June 1980.

DR D W BUSTARD
THE QUEEN'S UNIVERSITY OF BELFAST

PORTABLE PASCAL PLUS COMPILER
Nov 78 - Sept 84

Background

The objective of this project, which is funded through the EMR (q.v.) Contract mechanism, is to produce a compiler-interpreter system to allow Pascal Plus to be easily implemented on a variety of computers.

Achievements

A pre-release version of the portable-plus system was completed in 1980 and has still been installed on a range of different processors including LSI/PDP 11s (running RT-11 and RSX-11), Vax (running VMS), Z80 (running CP/M) and Motorola 68000.

Experience gained from this work led to the production of a revised system which is now available for distribution. In the new system Pascal-plus is aligned with the ISO Pascal Standard, keeping the full Pascal language as a pure sub-set [2]. The alignment produced a significant expansion in the size of the compiler which was counteracted by splitting it into two passes. Also for ease of partability the compiler is now coded in standard Pascal rather than Pascal-plus.

Enhancements due for release in December 1983 include a model p-code to assembly language translator [3,4] and a program development tool that assists with both the preparation of a program text and the detection and correction of compilation and logical errors [5].

For further information and details of availability, contact Dr Bustard.

References

1. J.Welsh and D.W.Bustard, 'Portable Pascal-plus', in *Proceedings of Pascal Experiences Conference*, Gothenburg, Sweden, June 1980.
2. D.W.Bustard and J.Welsh, 'A Tutorial Guide to Pascal-plus', Department of Computer Science, Queen's University, Belfast, 1982.
3. R.Orr, 'A Naive Pascal-plus P-code to Assembly Language Translator', Department of Computer Science, Queen's University, Belfast, 1982.
4. A.Rammasubu, 'A Study of the Optimisation of Portable Object Code', MSc Dissertation, Department of Computer Science, Queens's University, Belfast, 1982.
5. R.M.S.Young, 'A Pascal-plus Compiler for an Interactive Environment', MSc Dissertation, Department of Computer Science, Queen's University, Belfast, 1981.

Belfast (Queens University)

R M MCKEAG
THE QUEEN'S UNIVERSITY OF BELFAST

COLLABORATIVE RESEARCH INTO PARALLELISM
June 80 - Oct 81

Background

The investigator is interested in several aspects of parallelism, in both systems and applications programming. In the former he is concerned with operating systems problems, such as distribution and protection. In the latter he is concerned with the use of parallelism in the design of systems and with how a parallel design may have to be transformed into an implementation with a different process structure.

This grant provided funds for the investigator to make several short study visits to a number of centres in the UK and USA to examine current ideas on parallelism.

Achievements

The investigator visited the University of Southern California (Professor Brinch Hansen: Edison and Highly Concurrent Languages), the Argonne National Laboratory (Dr Boyle: Program Transformations; Dr Dritz: Parallelism in PL/1), Carnegie-Mellon University (Professors Wulf and Jones: Programming Multiprocessor Systems; Dr Joseph: Communication Systems) and the Massachusetts Institute of Technology (Professor Liskov: Distributed Computing).

Reports of the visits are available from R M McKeag.

PROFESSOR R M NEEDHAM
THE UNIVERSITY OF CAMBRIDGE

DISTRIBUTED COMPUTING USING WIDE BAND COMMUNICATIONS
DEVELOPMENTS OF THE CAMBRIDGE RING
Jan 80 - Dec 81
Aug 78 - July 81

These projects are now complete. They were concerned with the exploitation of modern local communications in the form of the Cambridge Ring in order to make organised use of a substantial number of machines. In particular, the cheapness of small machines poses the challenge of how to utilise them effectively.

The investigator set out to build an example of a distributed computing system that should be readily extensible, flexible, and controlled. This has been done, and though the DCS work is complete the development runs on in the Universe Project.

An extended account of the work is available in *The Cambridge Distributed Computing System* by R.M. Needham and A.J. Herbert, recently published by Addison-Wesley. That work contains a full bibliography of relevant publications.

The present system comprises about sixty heterogeneous machines and gives a daily service to a considerable number of research workers. It furnishes connections to various mainframes and to the outside world, and demonstrates how a number of different machines can work harmoniously together on the basis of a simple interconnection. Numerous industrial visitors, including those involved in the Universe collaboration, have discussed the project and we believe that good models have been provided to help industrial system designers.

Further information may be obtained from A.J. Herbert, A. Hopper or R.M.Needham at the Computer Laboratory, Corn Exchange Street, Cambridge CB2 3QG. Tel (0223) 352435.

DR F W BURTON and DR M R SLEEP
UNIVERSITY OF EAST ANGLIA

DISTRIBUTED EVALUATION OF APPLICATIVE PROGRAMS ON A HIGHLY INTERCONNECTED NETWORK
Aug 81 - July 83

Background and Nature of Research

Conventional programming languages are machine-oriented, requiring the programmer to tell the computer in considerable detail exactly what it must do to compute the desired result. Declarative languages such as SASL and PROLOG in principle allow the programmer to specify what is to be computed without going into any great detail about how the machine should do this.

This makes life easier for the programmer, but much harder for the implementor, who now has to work out the how from the what. The present situation appears to be that whilst declarative languages can lead to dramatic improvements in programmer productivity, current implementations are not well suited to high-performance applications.

Research Goals

Because evaluating expressions in a SASL-like language does not give rise to side-effects, sub-expressions may be evaluated in parallel. This suggests "buying speed" for such languages by employing large-scale hookups of single chip computers to perform parallel evaluation of sub-expressions. Programs written using the "divide and conquer" heuristic are particularly susceptible to this approach, giving potentially enormous parallelism. The goal of this project is to demonstrate, using simulation techniques, that a near linear speedup for divide and conquer algorithms is possible.

Achievements

Early ideas for achieving this goal have been published at various international conferences [1,4] and widely presented in seminar form. A sophisticated simulation vehicle with colour graphic instrumentation has been developed, and early (encouraging) simulation results presented at the MIT/ACM conference [6]. The simulator has been extended to simulate a distributed combinator reduction machine. A method for avoiding the quadratic growth in size of the code has been found [9]. Experiments with the programmable simulator confirm the main conjecture for suitable problems [11] although machine limitations restricted the number of processors to 24. An annotation approach to increasing user control has been developed [12]. In parallel with the architecture work, a new process-oriented formalism for modelling distributed evaluation mechanisms has been developed and the results are being presented in [7,8].

Work in Hand

Because it is already clear that the early approach can be made to work for some problems, considerable effort is being devoted to extending the basic results.

The departure of Dr F.W.Burton for Colorado has led to the early closure of this grant, but related investigations continue under a new DCS grant awarded to Dr Sleep.

Applicability

Tree algorithms are well-suited to the present architecture, which in principle at least can be constructed directly from emerging standard VLSI components.

Further Information

Contact Dr Sleep (0379 77288 or 0603 503660)

References

1. M.R.Sleep, 'Applicative Languages, Dataflow and Pure Combinatory Code', *IEEE COMPCON80 Digest*, San Francisco, February 1980.
2. M.R.Sleep and F.W.Burton, 'Towards a Zero Assignment Parallel Processor', *Proceedings 2nd International Conference on Distributed Computing Systems*, Paris, April 1981.
3. M.R.Sleep and F.W.Burton, 'Communication in a Distributed Implementation of an Applicative Language', *Proceedings 6th ACM ICS on Systems Architecture*, London, April 1981.
4. F.W.Burton and M.R.Sleep, 'The Zero Assignment Parallel Processor Project', *Symposium on Functional Languages and Computer Architectures*, Lab. for Programming Methodology, Chalmers University, Goteborg, July 1981.
5. Keller and M.R.Sleep, 'Applicative Caching', *Proceedings ACM/MIT conference on Functional Programming Languages and Computer Architectures*, Portsmouth, New Hampshire, October 1981.
6. F.W.Burton and M.R.Sleep, 'Executing Functional Programs on a Virtual Tree of Processors', ref. as in 5.
7. Kennaway and M.R.Sleep, 'Expressions as Processes', *Proceedings ACM symposium on LISP and Functional Programming*, Pittsburg, August 1982.
8. R. Kennaway and M.R. Sleep, 'Parallel Implementation of Functional Languages', *Proceedings 1982 International Conference on Parallel Processing*, Ohio, August 1982.
9. F.W.Burton, 'A Linear Space Translation of Functional Programs to Combinators', *Information Processing Letters*, Vol 14, No 5, pp 201-204, July 1982.
10. F.W.Burton, 'An Efficient Functional Implementation of FIFO Queues', *Information Processing Letters*, Vol 14, No 5, pp 205-206.
11. F.W.Burton and M.M.Huntbach, 'Virtual Tree Machines', to appear in *IEEE trans. on Computing*.
12. F.W.Burton, 'Annotations to Control Parallelism and Reduction Order in the Distributed Evaluation of Functional Programs', to appear in *TOPLAS*
13. F.W.Burton, 'A Distributed Evaluation Mechanism for Functional Programming', *Proceedings IEEE Med. Electrotechnical Conf*, Athens, May 1983.
14. J.R.Kennaway and M.R.Sleep, 'Applicative Objects as Processes', *Proceedings 3rd International Conference on Distributed Computing Systems*, Miami, October 1982.
15. M.R.Sleep and S.Holstrom, 'A Short Note Concerning the Lazy Reduction Rules for Append', *Software Practice and Experience*, December 1982.
16. J.R.Kennaway and M.R.Sleep, 'Novel Architectures for Declarative Languages', *Software and Microsystems*, Vol 2, No 3, June 1983.

DR R DOWSING
UNIVERSITY OF EAST ANGLIA

THE SPECIFICATION AND IMPLEMENTATION OF PROGRAMS ON A MULTIPROCESSOR
April 80 - Sept 83

Background and Nature of Research

Programming distributed computing systems is different in nature as well as in kind as compared with single processor von Neumann machines. The differences which are addressed in this project, concern the way in which programs are designed for a multiprocessor or distributed computer system and the level at which algorithms need to be described for efficient implementation on a number of different architectures.

In order to implement a design on a multiprocessor or distributed system it has to be *partitioned* into a number of parts and these parts *allocated* to physical units in the system. Exactly how this is done depends on the nature of the algorithm and the target system architecture. This research is aimed at producing algorithm descriptions which may be mapped onto a range of architectures.

Research Goals

Originally the aims of this project were to investigate the properties required of an operating system to run on a multiprocessor system, such as CYBA-M [1], to identify the hardware support necessary for such systems. Concurrently with this it was intended to study the design and specification of a number of programs on a multiprocessor to understand the design methodology and design aids required for distributed systems.

Due to a number of problems, including the move of the grantholder to another university and difficulty in recruiting personnel, it has proved impossible to tackle both of these tasks and the decision was taken to concentrate on the second objective, the design and specification of programs for distributed systems.

The aim of the project is to produce a distributed version of Path Pascal [2] on a Cambridge Ring based collection of multimicro computer systems and to measure the performance of a range of examples in this language. A subsidiary aim is to discover weaknesses in the language with a view to proposing modifications or a new language.

Achievements

As mentioned previously there have been problems in appointing staff and progress has been relatively slow. However, for the present calendar year work has progressed more rapidly, a number of internal reports have been produced and a number of seminars presented on the work. A new interpreter for Path Pascal has been designed and implemented requiring much less space than the previous version. This version runs under the IDRIS operating system. A study of a number of disserent multiprocessor algorithms for an assembler coded in Path Pascal has been completed including performance measurements on a single processor simulator. Some papers on this study are in preparation.

Work in Hand

The new interpreter system has to be ported to a multiprocessor Intel 8085 system. This is straightforward but time consuming. A paper study is being performed on a language based on path expressions suitable for implementation on a distributed system, for which Path Pascal is not suitable. Time will preclude and implementation of this language.

Other than the research team, the investigator and the two research assistants, a research student is investigating languages for describing the architecture of distributed computing systems and a research assistant on a contract funded by British Telecoms is investigating the design and specification requirements of telecommunication systems.

Applicability

The applicability of the project is demonstrated by the interest of British Telecoms and their funding of a research contract.

Further Information

May be obtained from the grantholder.

References

1. E.L.Dagless, 'A Multimicroprocessor - CYBA-M', *Proceedings of IFIP*, pages 843-9 (1977).
2. R.H.Campbell and R.B.Kolstad, 'Path Expression in Pascal', Dept of Computer Science report, University of Illinois (1979).

Various internal memos on the project are available from the grantholder.

DR M R SLEEP
UNIVERSITY OF EAST ANGLIA

THE POTENTIAL OF PURE COMBINATORY CODE FOR DISTRIBUTED COMPUTING
Jan 80 - Dec 82

Background and Nature of Research

Conventional computers do one thing at a time to elements of a large central memory, and most conventional programming languages reflect this fact: for example when writing a FORTRAN program which manipulates a large array of data, the basic assumption of every programmer and analyst is that examining or changing data elements involves roughly equal costs, where ever the elements are places in memory.

Because fifth generation diffusion architectures dynamically decompose a larger computation into a number of sub-computations which are distributed and independently executed on physically distinct VLSI components, a fundamental assumption of conventional computing fails. A new model of computation, in which account is taken of distributed costs, is required.

Research Goals

Because variable names must be looked up in an environment to give them a meaning, every use of a variable in a program implies communication costs which are (cleverly!) hidden in conventional implementations. Turner [1] reports the practicability of variable-free code, in which the detailed distribution paths are encoded using *combinators*. The goal of this project was to investigate the potential of such code for distributed implementations.

Achievements

This project started with a strong belief that combinatory code was of fundamental importance for distributed computing: early ideas are presented in [2]. It ends with a surprisingly simple picture of how combinators work, which shows clearly that key combinators are really just data-switching elements which can be used to avoid sending data to places it is not required. This "director string" model [3] emerged from work with Dr J R Kennaway who showed that combinator translations can in the worst case grow quadratically with the size of the original program [4]. A compile time balancing scheme was suggested, and details appeared in a paper by Dr F W Burton [5] which lead to a linear space translation. A simpler count-based encoding of director strings has been shown to yield a near linear translation [6].

Applicability

Direct use of these ideas could lead to and efficient parallel reduction machine based on combinatory code.

Contacts

Dr M R Sleep (0379 77 288 or 0603 503660).

References

1. D.A.Turner, 'A New Implementation Technique for Applicative Languages', *Software Practice and Experience*, January 1979.

2. M.R.Sleep, 'Applicative Languages, Dataflow and Pure Combinatory Code', *COMPCON 80 (Spring) digest*, San Francisco, February 1980.
3. J.R.Kennaway and M.R.Sleep, 'Director Strings as Combinators', to be published. Copies available from contact.
4. J.R.Kennaway, 'The Complexity of a Translation of Lambda-calculus to Combinators', to be published. Copies available from contact.
5. F.W.Burton, 'A Linear Space Translation of Functional Programs to Turner Combinators', *Information Processing Letters* July 1982.
6. J.R.Kennaway and M.R.Sleep, 'The Efficiency of Counting Director Strings', UEA CSD, copies available from contact.

DR M R SLEEP
UNIVERSITY OF EAST ANGLIA

PROGRAMMING AN EXTENSIBLE ARRAY OF TRANSPUTERS USING A NATURALLY EXTENDED FUNCTIONAL NOTATION
Oct 83 - Sept 86

Background and Nature of Research

Declarative programming using functional and logic notations is proving an increasingly serious contribution to Software Engineering, especially when inherent parallelism is exploited to yield usable performance. However, existing notations considerably restrict the users ability to express control information. Declarative code which runs with reasonable speed tends to make heavy use of knowledge of the underlying operational semantics of a particular implementation (eg. normal order for functional languages, depth first leftmost with cut for Prolog). The resulting code tends to be heavily obscured by such programming techniques.

Research Goals

The intention is to develop a *natural* extension of a functional language which accommodates process notations capable of expressing detailed control information. Early work in this direction is described in [1]. The method will be to work towards a soundly based formalism with clean semantics, and only then consider distributed implementation problems in detail. Nevertheless, the need for workable operational semantics will inform the design process throughout. Strong use of the *correspondence principle* will be made: in particular, the notation we seek should contain quite naturally the usual lambda calculus.

Achievements

This project has just started but it does so with the benefit of some early work funded by a previous grant. This has led to a new prototype notation (called DyNe) which is a significant advance over the earlier LNET [1] because there is exactly one central mechanism for abstraction.

Work in Hand

Various strands from the earlier project need to be written up, including director strings and a new implementation technique for functional languages. Because of the correspondence principle, we expect such work to inform the main task of developing DyNe.

Applicability

DyNe is seen as a very high level systems programming language for transputer networks, which contains the usual lambda calculus as a sub language.

Contacts

Dr M R Sleep 0379 77 288 or 0603 503660

References

1. J.R.Kennaway and M.R.Sleep, 'Expressions as Processes', *ACM Symposium on LISP and Functional Programming*, Pittsburg, August 1982.

DR G D PLOTKIN and A J R G MILNER
THE UNIVERSITY OF EDINBURGH

SEMANTICS OF A NON-DETERMINISTIC AND CONCURRENT COMPUTATION
Oct 81 - Sept 84

Background and Nature of Research

In the past five years, the focus of interest on distributed computing has led to tremendous interest in the design of concurrent programming languages, and in the need to describe and vigorously analyse the design of distributed hardware. Concurrency is now no longer regarded as something to be grafted on to the classical sequential view of computation; it is rather seen as a basic idea, of which sequential computation is just a special case.

Research Goals

Like its predecessor, this project - which began at full strength in 1982 - concentrates upon the central concepts both of the computations and of incomplete (hence non-deterministic) descriptions of computations. The aim remains to isolate these concepts, to find suitable mathematical expressions for them, and to develop techniques for analysing systems and languages in terms of them.

The present focus of the project is on the mathematics of non-determinism, and on the relation between algebraic calculi for concurrency (developed in part by the preceding project) and methods of proof based upon suitable logics - in particular, temporal and tense logics.

Work in Hand

1. Modal and Tense Logic

In previous work, Hennessy and Milner defined an equivalence between processes called observational equivalence [3,4]. They showed that this equivalence can be characterised by a modal language [3]. Colin Sterling has used this language to give complete modal proof theories for nondeterministic process languages [6].

Finer equivalences than observational equivalence have been defined between processes which take into account their liveness properties [2,5]. Joint work (between Colin Sterling and Matthew Hennessy) is in progress which has extended the modal characterisation result to these equivalences by using relativised branch time tense logics: logics where eventually operators are relativised to particular (general) paths into the future.

2. Operational semantics for "Liveness"

C Sterling and G Costa have been investigating a new way of handling the property of *liveness* (the assumption of unbounded but finite delay in computation) using variants of the operational semantic techniques employed by much work in Edinburgh projects. The problem arises when dealing with systems where (some) components evolve asynchronously. In modelling such systems, one usually considers languages in which there is a (binary) parallel operator. In the operational semantics for such languages, asynchrony is usually modelled by assuming rules which force interleaving of the actions of the operands of the parallel operator. The set of computation sequences produced by those rules then contains sequences which are "unfair". The aim is to find rules which will not produce unfair sequences, without enforcing synchrony. This has been achieved for a language which is a subset of Milner's CCS [1].

The subset of CCS considered does not allow a distinction to be made between weak and strong fairness, whereas full CCS does. Work is in progress which extends the previous results to these notations of fairness. Further work in progress is the development of "fair equivalences" between process which do not appeal to infinite computations.

Applicability

The research undertaken, however mathematically technical, is constantly guided by the need for good notations for expressing and analysing concurrent computations, and rigorous criteria by which to design articulate concurrent programming languages. Contacts with both ICL and STL (Harlow) have indicated that industry is interested in mathematical foundations, and is already willing to experiment with notations and techniques as they emerge.

References

1. G.Costa and C.Sterling, 'A Fair Calculus of Communicating Systems', *Foundations of Computation Theory*, 1983. *Lecture Notes in Computer Science*, Vol. 158, 1983, (and an extended version as internal report CSR-137-83).
2. M.Hennessy, 'Axiomatizing Finite Element Delay Operators', internal report CSR-124-82.
3. M.Hennessy and R.Milner, 'On Observing Nondeterminism and Concurrency', *Lecture Notes in Computer Science*, Vol. 85, pp. 299-309, 1980.
4. R.Milner, 'A Calculus of Communicating Systems', Lecture Notes in *Lecture Notes in Computer Science, Vol. 92, 1980.*
5. C.Sterling, 'A Proof Theoretical Characterization of Observational Equivalence', internal report CSR-132-83, (to appear in *FCT* and *TCS*, Bangalore, December 1983).

N H SHELNESS
EDINBURGH UNIVERSITY

AN ARCHITECTURE FOR A MULTIPLE COMPUTER SYSTEM
Oct 77 - Sept 80

Background

This project commenced in 1973. It sprang from a supposition common to a number of other projects in the programme, that there exists a broad class of computer applications for which hardware containing more than one processor (CPU) would form a more effective base than hardware containing only a single processor. Increased effectiveness can be measured in terms of improved reliability, throughput, response, ease of growth or economy, the relative weighting of which will depend on the nature of the application.

An early study of how these various improvements could be effective in a system containing more than 2 or 3 processors led rapidly to a conclusion that such systems would need to make only minimal use of shared common memory and that this being the case, a more effective solution was to dispense with shared memory and to employ an inter-machine communications sub-system in its stead. An immediate result of dispensing with shared memory is that an application has to be programmed differently. The application requires a structure during execution that allows the code and data required by a sub-computation to be firstly identified and secondly transferred between and assigned to particular processors and their private memories.

From 1974-76 the group, especially Dr Laim Casey, devoted its efforts to developing and evaluating through simulation a structure (the distributed domain model) that would satisfy this requirement. During 1977 and early 1978 a number of improvements to this model were developed and evaluated. This led to the elaboration of a new, simpler and more powerful structure (the segment flow model) based on a generalisation of the distributed domain model.

In parallel with this latter effort, the group approached the SERC to fund the construction of a multi-computer test-bed on which the proposed models could be implemented and evaluated by successfully constructing a number of realistic pilot applications.

Achievements

The test-bed which was installed in mid-1978, was built from Ferrant Argus 700 components. There are three identical computers or sites. Each site consists of an Argus 700 CPU which serves as the interface processor. The three sites are linked via the peripheral bus of each of the Argus 700Fs to a fast inter-site bus into which is also connected an Interdata 70 computer. This Interdata 70 which serves as a peripheral and maintenance processor for the entire system runs existing software and is linked through a high speed local network to standard peripherals and the Departments file processor.

The basic system was completed during 1980 but was still far from robust. However enough progress was made to produce some initial performance measures.

A segment transfer between two domains running on the same site took approximately 3 milliseconds of elapsed time. The same transfer between two domains running on different sites took approximately 25 milliseconds of elapsed time.

Edinburgh

References

1. L.Casey and N.H.Shelness, 'A Domain Structure for Distributed Computing Systems', In *ACM Operating Systems Review*, Vol 11, No 5 and CSR-8-77 Edinburgh Computer Science Dept.
2. L.Casey, 'Structures for Distributed Computing Systems', Internal Report CST-2-77, Edinburgh Computer Science Dept.
3. L.Casey, 'On Kernels for Distributed Computing Systems', Internal Report CSR-27-78, Edinburgh Computer Science Dept.

DR J M BACON
HATFIELD POLYTECHNIC

AN INVESTIGATION OF SOFTWARE STRUCTURES TO FACILITATE DISTRIBUTION
June 81 - May 83

Background and Nature of Research

This project is to explore the extension of well-established principles for the design of single-site operating systems and concurrent programming languages to local-memory based distributed systems.

Aims

To Establish a working design methodology for concurrent software. To evaluate alternative mechanisms which may be provided for concurrency, structuring and protection and to extend the associated system structures to accommodate a distributed system and to integrate communications handling into the overall design.

To study distributed resource allocation. To implement and evaluate simple, basic models on a small system rather than a full production system.

Achievements

A five-site, distributed system has been implemented based on a 'soft' token ring. Communications software follows the ISO reference model.

Two kernels have been implemented which provide asynchronous and synchronous message passing. Only the message passing primitives differ in the two kernels and process level software will run on both. Software to monitor communications traffic has been developed so that one site may be used to initiate and time evaluation tests running on other sites.

Tests have been carried out to measure the overhead of message buffering and consequent reduction in time available for running concurrent processes in such a system.

Work was started to simplify the communications handling software to exclude low level checking and retransmission and to measure the consequent increase in performance. This work has subsequently been transformed to a more realistic 'hard' ring based system.

Process level software for file service, printing service and device handling has been developed and has provided a basis for the study of the design of concurrent software and dynamic resource allocation in a distributed environment.

Applicability

The work relies heavily on MSc and BSc project work for implementation manpower, involving a number of sandwich and part-time students. A primary application of the system is educational.

For further information contact Jean Bacon.

DR L C W DIXON
NOC, HATFIELD POLYTECHNIC

THE ADVANTAGES OF A PARALLEL COMPUTATION FACILITY FOR GLOBAL OPTIMISATION
Jan 81 - Dec 83

Background and Nature of Research

With the advent of relatively cheap central processor units computer systems with many CPU's operating in parallel can be confidently expected to be readily available in the near future. The NOC is investigating how such parallel systems can be used to solve practical optimisation problems (in particular the global optimisation problem) more efficiently.

Research Goals

In our previous research contract (Development of Optimisation Algorithms for Parallel Computation, S E Hersom) it was shown that the solution of four distinct classes of optimisation problems ought to benefit by being undertaken on appropriate parallel systems. It is hoped to demonstrate the benefit on all four classes during this project.

In doing this we hope:

1. To investigate further the holistic principle developed at the NOC; applying it to a wider range of algorithms than currently achieved.
2. By considering the results of the implementations of holistic Torn & Price algorithms to improve the structure of these algorithms taking more account of the different type of computer on which they are being applied.
3. To investigate other ways of introducing parallelisation into optimisation algorithms particularly by investigating the possibility of (a) a parallel line search algorithm, or (b) the use of parallel line searches, (c) the construction of more complex function models given the new facility.

Achievements

During the year considerable use has been made of two parallel computing systems namely, the SERC DAP system at Queen Mary College, London University and the SERC NEPTUNE system at Loughborough University. The parallel Newton method has been successfully implemented on the DAP system showing considerable time savings and allowing research to be undertaken including parallel calculation of gradients and Hessians, a parallel line search algorithm and a 4-dimensional parallel sweep. All showed the expected gains in efficiency.

Following the success of the parallel Newton method, this work was extended to look at the solution of partial differential equations via finite elements and optimisation. This has now been successfully implemented and tested for linear systems and great time savings were possible using our parallel implementations.

By using the leastsquares finite element formulation any nonlinear partial differential equation can be solved by posing it as a nonlinear optimisation problem and applying a nonlinear preconditional conjugate gradient method. This approach maps ideally onto the DAP system and has been successfully implemented. The possibility of implementing parallel versions of Schlagen's method and of the Mexican tunnelling algorithm was investigated but abandoned when weaknesses were discovered in both.

Implementations of the holistic Price method have been successfully undertaken on both the DAP and NEPTUNE systems.

Work in Hand

The extension of the work on the parallel finite element/optimisation approach to nonlinear partial differential equations is in hand; a truncated Newton method is in course of implementation.

The people working on this project are: L C W Dixon, K D Patel (SERC Research Assistant) and P G Ducksbury (SERC Research Student).

Applicability

It is anticipated that the software developed under this project will later be as valuable to industry as the widely used codes currently available. In particular, of the codes so far investigated, it is anticipated that a parallel method for solving nonlinear partial differential equations of the type being developed will prove very valuable. Mr D Hunt of ICL has agreed to act as Industrial Supervisor for the above project being undertaken by P.G.Ducksbury.

Further information and copies of the references listed below can be obtained from L C W Dixon, Numerical Optimisation Centre, The Hatfield Polytechnic.

References

1. L.C.W.Dixon and K.Patel, 'The Place of Parallel Processing in Numerical Optimisation IV: Parallel Algorithms for Nonlinear Optimisation', NOC TR 125 Feb 1982. Presented at Symposium on Vector & Parallel Processors, organised by IBM Rome, March 1982.
2. P.G.Ducksbury, 'The Implementation of a Parallel Version of Price's Algorithm on an ICL DAP', NOC TR 127 May 1982.
3. K.Patel, 'Implementation of a Parallel (SIMD) Modified Newton Algorithm on the ICL DAP', Presented at Progress in the Use of Vector and Array Processors, Bristol University, August 1982.
4. K.Patel, 'Parallel Computation and Numerical Optimisation', NOC TR 129, Presented at the International Institute on Stochastis and Optimisation, University of Milan, September 1982.
5. L.C.W.Dixon, P.G.Ducksbury and P.Singh, 'A Parallel Version of the Conjugate Gradient Algorithm for Finite Element Problems', NOC TR 132, December 1982.
6. Papers presented at the *EURO VI Conference* in Vienna, August 1983; *The IFIP Symposium*, Copenhagen 1983; and *The IFIP Workshop*, Gargana 1983, will be available shortly.

Hatfield

DR J A GORDON
HATFIELD POLYTECHNIC

SECURE HIGH-SPEED DATA TRANSMISSION
March 83 - Feb 86

Aims

The aim of this programme is to develop and prove the strength of a cryptographically secure technique for data transmission at Megabit rates.

Background

At the present time there is a rapidly growing commercial awareness of the need for the protection of the privacy of data transmitted over public networks. The driving forces behind this interest may be traced to, among others, the following sources:

a. Public pressure, especially in the USA over the issue of personal privacy. This pressure has resulted from concern regarding the control of the flow of information between data bases set up to store medical, criminal, financial and other records on private citizens, and concern of the degree of vulnerability of such data in the hands of possibly unscrupulous computer personnel. These matters have recently been the subject of study of the Lindop Committee on Data Privacy in the UK.

b. The growing threat of computer crime whereby, for example, well informed but unauthorised groups of experts have been able to manipulate funds to their gain. Here they take advantage of the non-conservative nature of money, which is actually a form of information, and which may, contrary to Statute Book law be both created and destroyed - it being the single main task of our financial institutions to attempt, by imposition of accountancy practice, to force upon money a conservative property. In this area the notion of "stealing" data is particularly ironic.

c. The likelihood in the medium term future of the proliferation of various forms of electronic mail, and the need for public reassurance regarding the privacy of communication.

d. The growing tendency towards the dispersed, electronic society - a society in which business is conducted between organisations and individuals in widely separated geographical locations, who are nevertheless able to carry out coherent, coordinated tasks by means of sophisticated telecommunications systems.
The business community will no doubt be happier to see progress in this area if it can be assured of the security of its proprietary information against eavesdropping from rivals.

e. The explosive growth over the last decade of the use of computer networks.

f. The expected proliferation of point-of-sales terminals and electronic-funds-transfer.

g. The introduction by (for example) PTTs of specialised satellite facilities for business.

This interest in the need for data protection has had several results, among them:

a. In the USA the National Bureau of Standards has adopted a Data Encryption Standard (DES) known as FIPS-46 for the cryptographic protection of data. The use of this cipher has become mandatory for all (non-defence) Federal agencies.

b. The British Standards Institute is currently meeting to decide upon the attitude of the UK towards the standardisation of encryption techniques.

c. The International Standards Organisation has invited member countries to submit proposals for discussion regarding the standardisation of cryptographic techniques.

d. The academic community, especially in the USA has begun to take an interest into the information theoretic aspects of cryptography, and many novel techniques have appeared in recent issues of learned journals. Among these techniques are the various Public Key Crypto-systems, and various techniques for key exchange and for authentication.

Further Information

For further information contact Dr Gordon, Department of Electrical Engineering, Tel 070-72-68100, Ext 410.

DR K H BENNETT
UNIVERSITY OF KEELE

A DISTRIBUTED FILESTORE
Oct 79 - Sept 82

Background and Nature of Research

Data is a crucial asset for many organisations, and the reliable storage of data in computer systems is hence of considerable economic and practical importance. The purpose of this project has been to study how the replicated resources that occur in distributed systems can be harnessed to provide highly reliable file storage in a way that is simple for the user to exploit.

Research Goals

The original goals were to study a distributed filestore, by implementing a design, and using simulation and modelling to provide quantitative analysis. Two major objectives were set: firstly, to aim at a naming scheme which would hide location dependence; and secondly, to provide highly reliable storage through a single name of a user file, with the degree of reliability indicated by an integer.

Achievements

A working distributed filestore supporting reliability through multiple copy techniques has been demonstrated. The filestore is implemented across two LSI11/02 machines and is accessed from Terak and Microengine workstations; the interconnection is via a Cambridge Ring. A feature of the filestore is the "overlay mount" concept, to avoid interdependence of disc drive availability.

The Cambridge Ring has simulated several sub-projects, including simulation studies and real devices. A DMA access logic has been designed and simulation work on it is being undertaken; a terminal concentrator was designed early in the project and has been taken up elsewhere at Keele.

More recent work on the project, on majority consensus algorithms (including a working implementation) and network partitioning problems has been moved to a PDP11/34 running V7 UNIX; the LSI11/02 based implementation has pushed the modest equipment beyond its capabilities.

Some six external papers have been published; several more are in the pipeline. About 80 internal reports and 45 trip reports have been written; two workshops were held at Keele and research groups from a number of institutions have visited the project.

Work in Hand

The project was completed in December 1982.

Applicability

Some of the local area network results are directly applicable in products; the terminal concentrator is being used by the Keele Computer Centre for a TS29 PAD implementation. Informal discussions with several industrial concerns have occurred, although no direct exploitation has been made. It is expected that the distributed filestore algorithms could be incorporated into a UNIX-like distributed operating system.

Further Information

The grant-holder is Dr K H Bennett. The most useful references are as follows below (the first paper, which it is intended to publish externally, is obtainable from the above).

References

1. K.H.Bennett, O.P.Brereton, K.Lunn and P.Singleton, 'The Keele Distributed Filestore', Internal Report DCP/WD/80.
2. K.Lunn and K.H.Bennett, 'An Algorithm for Resource Location in a Distributed Computer Network', *ACM Operating Systems Review*, Vol. 15 No. 2, April 1981.
3. K.Lunn and K.H.Bennett, 'A Highly Reliable Distributed Directory System', *Proceedings of 2nd International Conference on Distributed Systems*, Paris 1981 (IEEE).

DR K H BENNETT
UNIVERSITY OF KEELE

A RELIABLE NETWORK-TRANSPARENT FILESTORE FOR A COLLECTION OF PERSONAL WORKSTATIONS
Dec 82 - Dec 83

Background and Nature of Research

The context of this project is broadly similar to that of the preceding research (q.v.). However, it is hoped to show that the ideas for a reliable filestore can be incorporated into an existing distributed operating system (UNIX United).

Research Goals

Firstly it was decided to undertake a study of filestores to examine possible future directions and problems. Secondly, work on the replicated files was to continue, using UNIX United as the research vehicle. It was planned to produce a working implementation of a majority consensus algorithm for managing replicated file consistency. Finally it was planned to start work on a collaborative venture with Prof. Randell's group a Newcastle, on network topology problems in UNIX United.

Achievements

The study of research problems in filestores led to the submission to SERC for funding to support a project "Structures for Reliable Filestores". This envisages filestores developing into reliable "object stores" which not only retain the state information but also procedures to manipulate that state. It is thought that this approach will provide a unified solution to the problems of replicated files, primitives for database implementation and mixed media files. Continuing the theme of reliability, it is also hoped to undertake work on the management of partitioning in distributed systems.

An implementation of a majority consensus on top of UNIX United is under way and is on target for completion by the end of the project.

A pre-release version of UNIX United was made available by Newcastle, and successfully mounted at Keele. Feedback to Newcastle has been provided.

Work on the network topology problem has started; a definition of the problem has been produced.

A number of papers have been produced for submission for external publication.

Work in Hand

Implementation of the majority consensus algorithm is continuing. This will form the basis for future work on replicate copy management. The research on network topology problems is getting under way and it is hoped to complete this study be October 1984.

Further work on the "object store" is being undertaken, with the objective of producing a specific design proposal document. It is planned to produce a very simple demonstration of mixed media files in order to obtain a "feel" for problems and applications.

Applicability

Initial discussions with industrial firms suggest that there is considerable interest in our approach to reliable media files.

Further Information

The grantholder is Dr K H Bennett. The most useful references are those given for the earlier project.

PROFESSOR P J BROWN and DR P H WELCH
UNIVERSITY OF KENT

COMPILING SERVERS FOR THE CAMBRIDGE RING
March 82 - Feb 85

Background and Nature of Research

The development of inexpensive but powerful micro-processors, together with high bandwidth communication technologies, has the potential for changing the ground rules for the design of computing systems. A network of small specialised machines may have several advantages over a large general purpose computer in trying to satisfy a variety of demands from a large number of users. Since each machine is physically separate, the choice of hardware and software for its realisation may be independent of what exists elsewhere in the network and dependent on the current "state-of-the-art". Software maintenance and upgrading on the separate machines will be simpler than on the traditional central mainframe (whose operating system may contain millions of lines of code). Extension of the network facilities may be made gradually and in line with user demand. Finally, because of the true parallelism with which the network services may be performed, the response time to user demand should be quicker.

Research Goals

This research is investigating the economics and feasibility of compiling servers on a network such as the Cambridge Ring [1]. The idea of a compiling server can best be understood by relating it to a printer server. The latter provides a specialised piece of equipment, a line printer, that can be employed by any users on the Ring when they want to do printing. Recently, specialised pieces of equipment have been built to do compiling of particular programming languages. Assume such a machine exists for compiling a language L. Then this machine could be connected to the Ring to provide an L compiling service for all Ring users. Instead of having several L compilers running on different machines on the Ring, there is this one L server. The advantages lie in standardisation and in performance. The specialised machine for compiling L should be better at doing this task than a general purpose machine.

This technique and its implementation in our environment is described in [2]. A similar approach may be used to provide other types of service, eg. text processing, and these will be investigated if time permits.

Achievements

A portable package (known as the "Kent Ring Handler" [3]) has been supplied to the SERC to be made available to other SERC sponsored research groups. This implements, in Pascal, all the Cambridge Ring protocols in common use at the University of Kent. The package was developed on a Pascal MICROENGINE (a trademark of the Western Digital Corporation) where it forms the basis for a "multi-server" [4] which includes general file transfer utilities and a preliminary compiler server. A report on the performance obtained and experience gained in implementing these protocols has been published [5]. Experimental compiler servers for Pascal and Hope have been built, and several other services provided.

Work in Hand

Work has been transferred from the MICROENGINE to the SAGE computer, in order to overcome certain problems with the former. Now this has been done, the project can begin to provide production services. An important aim of the project is to evaluate the practical usage of compiler servers, and the production servers will be used for this evaluation.

Applicability

The aim of this research is to develop techniques for the distribution of software services to micro-processors around a local area network and determine the effectiveness of the resulting system. If the results are positive there is no limit to the type of services which it might then be profitable to distribute.

Further Information

Contact Dr P H Welch, Computing Laboratory, University of Kent.

References

1. A.Hopper, 'The Cambridge Ring - a Local Network', *IEE Micro School* (Easter, 1980).
2. R.P.Bird, 'A Compiler Server Node in a Local Area Network', *Proceedings of International Computing Symposium*, B.G. Teubner, Stuttgart, 1983.
3. T.E.Schutt, 'Ring Handler [1.0] - User Manual' University of Kent (1981).
4. T.E.Schutt and Dr P.H.Welch, 'Applying Micro-Computers in a Local Area Network', *Conference Proceedings, Local Networks and Distributed Office Systems*, Online Publications Ltd, (1981).
5. T.E.Shutt and Dr P.H.Welch, 'Byte Streams on a Micro-Computer', *Conference Proceedings, Struktur und Betrieb von Rechensystemen*, NTG/GI-Fachtagung, VDE-Verlag GmbH (March, 1982).

MRS H BROWN, S E BINNS and D J CAUL
THE UNIVERSITY OF KENT

TYPESETTING AND TEXT PROCESSING SERVER FOR THE CAMBRIDGE RING
Jan 82 - Jan 85

Background and Nature of Research

Computer document preparation and typesetting systems have grown up in two separate worlds. In the publishing world the emphasis is on quality and flexibility, but systems are typically tied to a particular typesetter. In the general computing and office world the emphasis is more on ease of creation and updating, but output is restricted to single-font lineprinters or daisywheel printers. Hardware developments are such that specialised typesetting systems are becoming almost indistinguishable from general purpose computer systems equipped with a high quality printing device. Thus it should now be possible to bring the best of the two worlds together to create *high quality* document preparation systems which are not tied to a particular typesetter.

Research Goals

The initial aim of the project is to concentrate all the necessary specialised hardware and software into a "typesetting server" for the Cambridge Ring - thus providing a high quality service to a large community of users at a moderate cost. The long term aims are to investigate the best way to integrate and such a server into a distributed system, and to develop more interactive and user-friendly typesetting systems.

Only systems producing "device-independent" output will be implemented or developed. This will ensure that documents can be created or previewed on a graphics screen before printing and that a higher resolution printer or typesetter can be incorporated into the server.

Achievements

The initial version of the server - based on an ICL PERQ and a Canon LBP-10 laser printer (240 dots/inch) - has been set up. The Canon LBP-10 has been interfaces to the Cambridge Ring via an Orbis 68000 which acts as an intelligent controller in DTF via the ring.

A preliminary version of Professor Knuth's TEX typesetting system has been implemented on the PERQ; TEX output can be previewed on the PERQ screen and printed on the Canon LBP-10. In addition, the new "typesetter-independent" version of the UNIX troff has been implemented on a VAX 11/780. An experimental troff-to-Canon service with 30 users is in operation.

Work in Hand

The server is currently being integrated properly into the ring. This includes providing better error reporting and recovery facilities, and making use of a file server on the ring to hold font libraries.

At the moment, the preliminary version of TEX runs under the POS system on the PERQ. As soon as PERQ UNIX with Pascal becomes available TEX will be re-implemented and integrated properly into the server. Work is also under way to allow output from both TEX and troff to be printed on a Monotype Lasercomp.

Applicability

Improved techniques for document preparation are clearly needed for office systems, for in-house printing facilities, and for interfacing office systems with publishers systems.

Further Information

Contact Heather Brown, Computing Laboratory, University of Kent.

DR F K HANNA
UNIVERSITY OF KENT

DISTRIBUTED COMPUTING SYSTEMS FOR INTERACTIVE KNOWLEDGE BASES
Oct 78 - Sept 81

Background

One practical problem in the development of artificial intelligence systems over the past twenty years has been the immense amount of processing power necessary to implement relatively simple systems. Since computer hardware, including processors, is not relatively cheap, it seems worthwhile considering the use of coupled microprocessors to meet the need for intensive computation. This project has been exploring such an approach, both theoretically and practically. Since the particular interest was the possibility of coupling large numbers of processors (eg several hundred), it was necessary to impose the further requirement that the processors be loosely-coupled, so that arbitrarily many units could be connected without introduction problems of hardware memory access. The intention was to develop general principles as far as possible, rather than simply constructing a single ad hoc system.

Achievements

The first stage of the project (October 1977 to August 1980) involved the design and implementation (hardware and software) of a multiprocessor, multiprocessing system based on Motorola 6800s. The system included three loosely-coupled processors, each containing an identical list-oriented multi-processing interpreter, with a fourth processor to handle data communications. Interprocess communication was effected by passing arguments to procedures in remote environments, and there were facilities for parallel evaluation of arguments and for parallel execution of statements [1]. This implementation demonstrated that all these ideas could be combined in a working system, but the result was not particularly suitable for large knowledge representation applications, owing to memory limitations.

The second stage of the project began in July 1979. It has now been decided that any further implementations were to be carried out on DEC LSI-11s, and so a revised implementation was required.

No significant problems of a practical nature were encountered in either implementation, leading to the conclusion that the main issue involved constructing distributed knowledge representation systems lay in the efficiency of the resulting programs dynamic behaviour. It was not a problem to implement a system on multiple processors, but it might be difficult to do so efficiently; moreover, it might be very difficult to prove (even informally) that a distributed implementation was efficient.

This led to a detailed survey of recent artificial intelligence work, in order to find either a suitably substantial test program for the knowledge representation language, or some theoretical basis for proving mathematical results (about the efficiency of possible implementations). Unfortunately, there is a severe lack, within artificial intelligence, of medium-sized, portable, documented, knowledge-representation programs which actually work. The conclusion of this part of the investigation was that most "knowledge representation systems" are in fact large, arbitrary, fragile programs which embody few theoretical principles.

Another area examined was that of database models, since it seemed likely that there might be some connections between distributed database work and distributed knowledge representation. In fact, database models seemed to be better defined than artificial intelligence notations, and they differ only slightly in the range of facilities they attempt to provide [5]. Since most artificial intelligence formalisms are not specified in database terms, it is difficult to translate database techniques directly for use in artificial intelligence programming, but in principle this approach should be possible. Many of the distribution techniques (eg for maintaining consistency) could then be applied directly to distributed knowledge representation programs.

The issues involved in knowledge representation are so ill-understood at present that it is unrealistic to consider distributing existing systems, since they are generally ad hoc programs. Particular applications in which there is a natural multi-dimensional structure (eg a distributed formation net, as in the large project at Carnegie-Mellon University) may lend themselves to distributed implementations, but in the area examined in this project (general "semantic network" representations of knowledge) the problem reduces to the less specific issue of how to implement a distributed list-processing system. Even then, the question of a realistic "bench mark" for such a system is difficult to resolve, in the absence of suitable test programs.

References

1. G.G.Coghill, 'Pleiades - A Multi-microprocessor System for Knowledge Manipulation', PhD Thesis, University of Kent, 1980.
2. F.K.Hanna and G.D.Ritchie, 'AM: A Case Study in AI Methodology', Internal report, University of Kent.
3. F.K.Hanna and G.D.Ritchie, 'Time-Efficient Implementation of Data Structures using Multiple Processors', Internal report, University of Kent.
4. G.D.Ritchie and F.K.Hanna, 'Semantic Networks - A General Definition and a Survey', Internal report, University of Kent.
5. G.D.Ritchie and F.K.Hanna, 'Structural Models in Databases and in Artificial Intelligence', Internal report, University of Kent.

PROFESSOR D A TURNER
UNIVERSITY OF KENT

A DENOTATIONAL LANGUAGE FOR DATAFLOW MACHINES
Mar 80 - Mar 82

Denotational languages are very high level programming languages without any explicit control structure - the two main groups of languages of this class are the logic languages (such as PROLOG) and functional languages. The investigator has been the designer of the functional language, SASL [1], and the purpose of the project has been to investigate the extent to which this language could be adapted to be the main programming language for a parallel computer architecture, such as a dataflow machine. The work was begun in liaison with Ronan Sleep of East Anglia, who adopted SASL for his work on parallel architectures [2]. The funds awarded were primarily travel money, to enable the investigator to visit other sites both in the UK and abroad (primarily the U.S.) to exchange ideas and to pursue the further development of the SASL language.

A great deal of progress has been made. First there have been some improvements in the underlying implementation technique (combinator reduction) that should ease its adaptation to parallel hardware. Secondly the investigator has developed a successor language to SASL, called KRC [3], which incorporates ideas based on set theory which greatly increase its programming power. During the past year, several papers were published by the investigator describing various aspects of the research [4,5,6]. A paper has also been prepared giving a fully formal proof of the correctness of the SASL compiler, and this is awaiting publication.

A major follow-up project has now begun, involving the construction at Kent of a large microprogrammed combinator reduction machine and the development on it of a complete operating system written in a functional language.

References

1. D.A.Turner, 'SASL Language Manual', St. Andrews University, December 1976.
2. M.R.Sleep, 'Applicative Languages, Dataflow and Pure Combinatory Code', *IEEE 20th Spring Computer Conference*, San Francisco, 1980.
3. D.A.Turner, 'KRC Manual', University of Kent, in preparation.
4. D.A.Turner, 'Prospects for Non-Procedural and Dataflow Languages', Infotech State of the Art Report, London, June 1981.
5. eds J.Darlington, P.Henderson and D.A.Turner, 'Recursion Equations as a Programming Language', in *Functional Programming and its Applications*, Cambridge University Press, February 1982.
6. ed Neel 'Functional Programs and Proofs of Program Correctness', in *Tools and Notions for Programming Construction*, Cambridge University Press, August 1982.
6. ed. Neel, 'Functional Programs and Proofs of Program Correctness', *Tools and Notations for Program Construction*, Cambridge University Press, August 1982.
7. G.M.Holms, 'Edit - As an Application of Functional Programming', University of Kent M.Sc. Thesis, October 1982.
7. S.Meira, 'Sorting Algorithms Implemented in a Functional Programming System', University of Kent Technical Report, June 1982.

PROFESSOR D A TURNER
UNIVERSITY OF KENT

AN APPLICATIVE LANGUAGE MACHINE AND OPERATING SYSTEM
Apr 83 - Mar 86

Applicative (or functional) languages are very high level languages without any explicit control structure. They are related to logic languages, such as PROLOG, but differ in being based on functions rather than relations and in being higher order (so functions may be passed as parameters or returned as results). Their potential future importance to the software industry lies in the fact that applicative programs can be much shorter and easier to reason about than conventional ones and also in their capacity for parallel execution which should make them a natural fit for novel highly concurrent architectures. The investigator has played a leading role in the development of applicative languages and their implementations [1,2,3] and has received SERC support under the DCS programme since 1980.

The purpose of the current project is to demonstrate the practical viability of applicative languages for systems programming by producing a complete multi-user operating system, with a UNIX-style filing system and a variety of production programs, all written entirely in an applicative language. This has never been done before and will involve overcoming a number of theoretical and practical problems. The work is being carried out on an ORION supermini computer, supplied by High Level Hardware of Oxford, which is being microprogrammed for the project as a combinator reduction machine [2] (an abstract machine designed to support applicative languages).

The project team consists of the investigator, two research associates, and several postgraduate students. So far an editor and a compiler have been written, and work carried out on the complexity of functional programs [4,5].

There is a developing interest in industry in practical applications of functional programming and the investigator is involved in regular discussions and exchange of information with a number of industrial firms including Burroughs Corporation, ICL, and L M Ericsson (the telecommunications company).

For further information contact the investigator.

References

1. D.A.Turner 'SASL Language Manual', St Andrews University, December 1976.
2. D.A.Turner, 'A New Implementation Technique for Applicative Languages', *Software Practice and Experience*, January 1979.
3. (eds) J.Darlington, P.Henderson and D.A.Turner, *'Functional Programming and its Applications'*, Cambridge University Press, February 1982.
4. G.M.Holmes, 'Edit - An Application of Functional Programming', University of Kent M.Sc Thesis, October 1982.
5. S.Meira, 'The Complexity of Sorting Algorithms Implemented in a Functional Programming System', University of Kent Technical Report, June 1982.

Lancaster

D HUTCHISON and W D SHEPHERD
THE UNIVERSITY OF LANCASTER

DIRECT COMPARISON OF RING AND ETHERNET TYPE SYSTEMS
Sept 81 - Sept 84

The purpose of this research is to investigate the relative performances of Ring and Ethernet-like systems over a wide range of different operating conditions using a local computer network which can be implemented by means of either a Cambridge Ring or Strathnet (an Ethernet-like system).

The project started in September 1981 and the research assistant was appointed in October 1981. The Cambridge Ring was installed in February 1982. The major part of the work since then has been to produce a working system which would provide a suitable vehicle for the performance investigations. This has involved the construction of a file server based on an LSI-11/23 with a Winchester disc, the design of interprocess communication primitives between the various computers attached to the network, the design and implementation of a traffic monitor to enable meaningful statistics to be gathered from the network, and the implementation of the basic block drivers for the Polynet DMA Interface to the PDP-11s. In addition the Strathnet Access units are being modified to interface to the second port on the Polynet DMA Interface to allow the operation of a dual system (Ring or Ethernet-like). It is hoped that the dual system should be operational early in 1983.

As a part of the programme a DEC LSI-11/23 and a 10MByte Winchester disc were purchased and four LSI-11/02 systems were given by DEC. Two of the four LSI-11/02 systems are being attached to the Cambridge Ring and the others to Strathnet. They will allow the investigators to look at the performance of personal machines attached to the two systems.

Over the past two years a number of papers have been given related to the work in progress and these are listed at the end of the report.

The past year has been mainly spent in setting up a dual system, both hardware and software, on which meaningful comparison measurements can be carried out. The first of these will be comparing the performance of our remote procedure call implementation across the Ring and Ethernet-like system respectively. Some preliminary work has been done in this area [8]. Digital Equipment Corporation (DEC) have shown interest in out work and are supporting a CASE student from October to look at the performance of their DEC Ethernet. This will enable us to compare the Ring against a commercially available Ethernet. In October the research group moved to the University of Lancaster.

References

1. G.S.Blair, D.Hutchison and W.D.Shepherd, 'MIMAS - a Network Operating System for Strathnet', *3rd International Conference on Distributed Systems*, Florida, October 1982.
2. G.S.Blair, J.A.Mariani and W.D.Shepherd, 'A Practical Extension to UNIX for Interprocess Communication', *Software Practice and Experience*, 13, 45-48 (1983).
3. W.D.Shepherd, G.S.Blair and D.Hutchison, 'Comparison of an Ethernet-like Communication System with the Cambridge Ring', *Proceedings IEE*, Vol. 129, Part E, No. 4, July 1982.

4. D.Hutchison and W.D.Shepherd, 'An LSI-11 Based Fileserver', *DECUS UK Conference*, University of Keele, March 1982.

5. D.Hutchison and P.Corcoran, 'A Microprocessor-based Local Network Access Unit', *Microprocessors and Microsystems*, Vol. 6, No. 1, Jan/Feb 1982.

6. G.S.Blair and W.D.Shepherd, 'A Performance Comparison of Ethernet and the Cambridge Digital Communication Ring', *Computer Networks*, Vol. 6, 1982.

7. D.Hutchison and W.D.Shepherd, 'Strathnet - A Local Area Network', *Software and Microsystems*, Vol. 1, No. 1, October 1981.

8. G.S.Blair, D.Hutchison and W.D.Shepherd, 'Implementation of a Local Network Operating System', *Localnet'83*, London, March 1983, Online Publications, pp 387-398.

9. S.J.Muir, D.Hutchison and W.D.Shepherd, 'ARCA - A Network Fileserver', to appear in the *Computer Journal*

10. J.R.Nicol, G.S.Blair and W.D.Shepherd, 'CRS - A Powerful Primitive for Resource Sharing in UNIX', submitted to *Software Practice and Experience*.

London (Imperial College)

S ABRAMSKY
IMPERIAL COLLEGE

AN APPLICATIVE PROGRAMMING METHODOLOGY FOR CONCURRENT AND DISTRIBUTED SYSTEMS
Oct 82 - Oct 84

Background and Nature of Research

There has recently been an upsurge of interest in applicative (or functional) programming. On one hand, applicative programs are simpler and more amenable to rigorous Software Engineering disciplines than their imperative counterparts; but in addition, they provide a more promising basis for the development of highly parallel architectures, since they are less wedded to a sequential model of execution. However, to date applicative programming has only been applied to concurrent and distributed systems, such as operating systems and networks of personal computers, in a limited and speculative way.

Research Goals

The proposed research has three main aims:

1. To develop a coherent and systematic style of applicative programming for concurrent systems.
2. To develop rigorous workable methods for formal specification, verification and transformation of such programs.
3. To investigate implementation techniques for applicative multiprogramming, in order to construct experimental systems, and to obtain some insight into performance requirements and feasibility.

Achievements and Work in Hand

The project has made very encouraging progress in this period. The practical work carried out by Richard Sykes, has achieved the following:

1. A complete implementation of the SECD-M machine [1] on the Perq (in Pascal), incorporating lazy evaluation, multiple processes, non-determinism in the form of fair merge (a significant extension of the original design), and I/O device interfacing, including multiple virtual keyboards and screens, character and S-expression disc I/O, and the Perq puck.
2. A LISPKIT compiler extended to incorporate merge as a primitive.
3. The design and implementation of a configuration language, which allows
    a. Compiled code to be loaded from a file and 'linked-in' to a system by connecting its input streams to output and input streams of the system.

    b. Device interfacing to be set up by mapping input and output streams of the system to input and output devices.
4. A number of applicative concurrent systems have been programmed, and run on a test-bed, including:

    a. A window manager, to allow multi-window I/O on the Perq display.

    b. Line protocols - some simple communications protocols programmed functionally - this work was done in collaboration with Colin Low.

c. A directory based filing system, including functionally programmed disc I/O - still being developed.

d. A Unix-like shell, supporting executable files I/O redirections to files, windows, etc. - still being developed.

e. A mouse-controlled line drawing program.

Some of this work is reported in [4].

On the theoretical side, some promising advances have been made in the semantics of applicative multiprogramming, including:

1. The development of a fixed-point semantics based on a categorical powerdomain construction which is an exact agreement with operational semantics, and is continuous even for unbounded nondeterminism.
2. The characterisation of classical powerdomain semantics by "indistinguishability modulo finite experiments", and the development of a hierarchy of non-deterministic languages and tests by varying the finiteness restrictions.

These results will appear in [2] and [3] respectively.

Applicability

The results are applicable to the development of functional programming as a realistic candidate for large-scale use. They also contribute to the development of high-level descriptions of concurrent systems. Finally, they provide some valuable information on implementation and performance requirements for applicative multiprogramming systems.

Further Information

Contact Samson Abramsky, Department of Computing, Imperial College.

References

Note: QMC CSL nnn = Queen Mary College Computer Systems Laboratory Report number nnn.

1. S.Abramsky, 'SECD-M: A Virtual Machine for Applicative Multiprogramming', QMC CSL 322, 1982.
2. S.Abramsky, 'On Semantic Foundations for Applicative Multiprogramming', to appear in *Proceedings of ICALP '83*, (to be published by Springer in the *Lecture Notes in Computer Science* series), also QMC CSL 321, 1982.
3. S.Abramsky, 'Experiments, Powerdomains and Fully Abstract Models for Applicative Multiprogramming', to appear in *Proceedings of FCT '83*, (to be published by Springer in the *Lecture Notes in Computer Science* series), 1983.
4. R.Sykes, 'Techniques for Applicative Multiprogramming', submitted to *ACM Conference on Principles of Programming Languages*, 1983.

R J CUNNINGHAM and J KRAMER
IMPERIAL COLLEGE

COMPUTER ASSISTED METHODS FOR DEVELOPING VERIFIED SOFTWARE
Jan 82 - Dec 84

Background

The aim of this research project is to investigate and develop software tools which will assist the specification, design and production of verified software for distributed systems. Origins of the work lie in a method developed by the investigators for applying proof-directed methods of programming to the design of hierarchical distributed processing systems [1,7,2]. This indicated a potential benefit from software tools which would support a constructive approach to the development of sound software.

Goals

The main thrust of the project is in the provision of an experimental environment which will assist a software engineer in verifying each phase of the software development process; getting a formal specification from informal requirements and pursuing a process of verifiable design to produce a sound maintainable produce. A facility for creating and updating a database of specifications is an important part of the proposed environment. Software tools based on formal logics are intended to be available for automating key steps in validating specifications and constructing verifiable designs. It has always been intended that the environment should be based on a Unix system and exploit other high-level software in order to develop new tools and maintain the specification database. The integrated system is intended to be a vehicle for experimental applications in the use of formal design tools for distributed systems.

Achievements

In considering the most appropriate route for obtaining sophisticated tools for formal analysis it was decided to exploit theorem proving expertise within the department [6] and use Prolog as a prototyping language. Some separation of technical issues in logic and algebra was also resolved.

In previous reports there have been descriptions of work on basic tools for the analysis of standard form logic [3], and research into rewriting rules for simplifying expressions in partial algebras [4].

A parser and intermediate form for an experimental specification language have been developed.

Work in Hand

A more powerful theorem proving module based on Connection Graphs [6] is being developed. A report is in preparation.

Software tools previously provided in Waterloo Prolog have been converted to Edinburgh Prolog for the purposes of portability and integration with the Unix system.

Work on a polymorphic type checker for the specification language has been initiated. Related Activity

Members of the project team collaborated in the organisation, lectures and demonstration at the BCS-FACS/SERC workshop on program specification and verification at York.

R.J.Cunningham participated by invitation in the ACM March workshop on high-level debugging and has also served on Validation, Verification and Testing Group of the DoI Software Tools Study.

Dr J.Kramer has contributed at an EWICS meeting on Application-Oriented Specifications and the SERC/DCS June Conference at Brighton.

A paper by Goldsack and Cunningham is included in the Pergamon Infotec State-of-the-Art Report on Software Engineering Developments [5].

References

1. R.J.Cunningham and J.Kramer, 'An Approach to the Design of Distributed Computer Control Software', *Proc. IEE Conference on Distributed Control Systems* 1977.
2. R.J.Cunningham and J.Kramer, 'The Design of Stable Distributed Processing Systems', Research Report 80/10, Department of Computing, Imperial College.
3. R.J.Cunningham and Silvana Zappacosta-Amoldi, 'Software Tools for First Order Logic', Research Report 82/19, Department of Computing, Imperial College. (to be published in *Software Practice and Experience)*.
4. A.J.J.Dick and R.J.Cunningham, 'Rewrite on Lattice-Structured Value Spaces', Research Report 83/7, Department of Computing, Imperial College.
5. S.J.Goldsack and R.J.Cunningham, 'Systematic Design Methods in the Design of Computer Programs', in *Software Engineering and Developments: State-of-the-Art Report 11:3*, (Editor P.J.L.Wallis), Pergamon Infotec 1983.
6. R.A.Kowalski, 'A Proof Procedure based on Connection Graphs', *JACM* 22 4, 1975.
7. J.Kramer and R.J.Cunningham, 'Towards a Notation for the Functional Design of Distributed Processing Systems', *Proc. IEEE/ACM International Conference on Parallel Processing*, Michigan 1978.

DR J DARLINGTON
IMPERIAL COLLEGE

DEVELOPMENT OF A GENERAL PURPOSE APPLICATIVE LANGUAGE MACHINE AND APPLICATIVE PROGRAMMING ENVIRONMENT
Oct 82 - Sept 85

Background

This project to develop software and hardware components of an applicative language programming system, started to receive DCS funding in October 1982. The work was previously supported by other SERC grants. The DCS funding allowed expansion of the work, in particular the construction of a parallel graph reduction machine, ALICE (Applicative Language Idealised Engine).

Goals

The aim of the project is to investigate the potential of the applicative languages for improving software reliability and efficiency. The mathematical origins of these languages mean that they inherit all the essential attributes of mathematical notations such as conciseness, comprehensibility and manipulability, in marked contrast to the more conventional procedural languages whose close attachment to the von-Neumann model of computation embodied in all present day machines precludes these advantages. In particular the project is investigating the development of the languages themselves, their application to real world problems, the use of formally based program development systems and implementations on highly parallel architectures.

Progress in 1982/83

Work proceeded along the planned routes at a satisfactory pace and is on target for an ALICE prototype in 1985. Highlights of the first year's activity were:

(i) The overall design of ALICE was simplified and carried a stage nearer concrete realisation. Final coding of the machine in Occam was started and was half completed as of October 1983.

(ii) The design of the network chip was completed. This 4 x 4 crossbar switch is asynchronous and self-timing and enables fault detection on the fly chip replacement in delta networks. The fabrication was planned for October 1983. A patent application has been filed via BTG.

(iii) Various alternative network topologies were investigated and a new class of networks developed, in many ways superior to delta networks.

(iv) A compiler for full standard LISP to ALICE CTL was developed.

(v) The OR-parallel Prolog interpreter was extended to incorporate AND parallelism.

(vi) A method of extending functional languages to incorporate unification was developed and is being studied. This enables one to incorporate facilities found in logic languages such as Prolog into a functional language.

(vii) The Hope in Hope compiler was further developed and optimised.

(viii) The Hope Pascal interpreter was further developed and distributed to a number of sites.

(ix) A range of application software was developed in Hope including a picture description package and a language independent structure editor.

(x) Applications of declarative languages to 'real time' problems were investigated and a methodology developed to allow necessary sequencing to be introduced in declarative notations in a systematic manner.

(ix) The design of a simple single user operating system for the ALICE prototype was started.

Papers Produced

1. J.Darlington, 'Unifying Functional and Logic Languages', Internal Note, Department of Computing, Imperial College.

2. J.Darlington, 'From Specification to Program via Transformation', Invited Paper Mathematical Centre Symposium on *Program Specification* November 1983.

3. J.Darlington and M.J.Reeve, 'ALICE and the Parallel Evaluation of Logic Programs', Invited Paper *Computer Architecture Symposium*, Stockholm, June 1983.

4. A.J.Field, 'A Structure Independent Switching System with System Level Fault Tolerance', Internal Report, Department of Computing, Imperial College, 1983.

5. M.J.Reeve, 'The ALICE Compiler Target Language Reference Manual', Internal Report, Department of Computing, Imperial College.

London (Imperial College)

DR M SLOMAN and DR J KRAMER
IMPERIAL COLLEGE

SPECIFICATION AND ON-LINE MANAGEMENT FOR DISTRIBUTED REAL-TIME SYSTEMS
Jan 83 - Dec 85

Background and Nature of Research

This project investigates specification techniques and on-line configuration management for distributed real-time systems. It is based on the CONIC project at Imperial College. The CONIC Architecture for a DCCS [1,3] is the outcome of a project funded by the National Coal Board on the use of microcomputers for monitoring and control in coal mines. Dr Sloman has an associated project funded by SERC on Communication Techniques for Distributed Control.

CONIC provides an integrated set of techniques and tools for constructing and managing large Distributed Computer Control Systems. The CONIC software architecture provides a two level language: one for programming individual software components (module definitions), and the other for the configuration of a distributed system from instances of these modules.

Future distributed systems will consist of large numbers of microprocessor based stations connected to a communication system. The complexity of structuring and controlling the hardware and software components in such large distributed systems requires the use of computer based tools for management of the system. Conventional system builders use a system description and generate an initial system, possibly with some checking for consistency with respect to the specification of the system structure.

However most systems are not static, but evolve as requirements change. Hence it must be possible to modify or extend parts of the system. Since it may not be safe or economic to shut down the entire system, it must be possible to introduce these changes while the system is running. Thus the configuration facility needs to be on-line, to have the capability of controlling individual components (eg. add, remove, start or stop) and must maintain information on the current status of the system. CONIC provides a suitable software component and the run-time environment to instantiate and interconnect these components dynamically, but tools are needed to check that changes are made in a safe and consistent way. A configuration language is necessary to allow the designer to express the logical structure of his system in terms of interconnected groups of modules, and the mapping onto hardware structure of stations and subnets. Ideally the system must be capable of automatically determining the physical interconnection structure of the hardware components but the languages should also allow this to be expressed by the designer. The language used to describe incremental changes should be the same as that used for initial system descriptions.

The configuration manager will have to provide a number of users (eg. plant manager, maintenance engineer, control engineer, plant operator) with information on the current structure and status of the system, and these users may have different views of the system. This indicates that different names or naming trees for objects may have to be supported. To be investigated are the issues of how groups of modules can be named and treated as a single entity to hide complexity or to reflect application structure (eg. subsystems). The configuration manager will have to maintain a database which contains all the above information and may have to provide different user profiles or access filters.

Individual Module Behaviour - A number of the more recent programming languages have recognised the need to separate the interface specification from the operational details given in the body of the package, or module. This has gone some way to providing a separation of specification from implementation information, but so far has only provided structural information. A major concern is that these interface specifications do not include behavioural aspects. In the same way as the algebraic approach has provided a sound basis for the specification of the behaviour of abstract data types, so there is a need for behavioural specifications, including real-time constraints, of message passing modules in distributed systems.

Composition of Modules - Using a suitable configuration language CONIC modules can be interconnected to form a system. These connections would be checked to ensure structural compatibility (ie. that exit to entryport links and have the same message types) and so to provide some security against erroneous configurations. The system description, together with rules of composition of individual module behaviours, would provide a means for specification of system behaviour. Behavioural compatibility could then be checked at the same time as structural compatibility.

CONIC also supports the notion of reusable modules by instantiation with different actual parameters. The affect of these parameters on module instance to its environment should be checkable. The on-line system configuration facilities discussed in the previous section will provide a powerful but potentially dangerous environment in which modules can be replaced, added or deleted at run time. Behavioural specifications with composition rules would help to ensure that only 'safe' modifications are made to the system.

The need, then, is for a practical, soundly based approach to the specification and composition of the message passing modules.

Work in Hand

Some progress in the configuration management area has been made. An initial off-line network builder which creates static configurations of CONIC modules for a network of stations has been implemented. [2] provides a model of the system configuration process and a framework for examining dynamic configuration. This proposes a language which specifies the logical configuration of nested groups of modules and provides the means for specifying changes to the initial configuration. Considerable work is still needed to refine these proposals and in particular to provide an elegant means to specify the mapping of the logical configuration onto a physical one.

Further Information

Contact Dr M Sloman or Dr J Kramer, Dept of Computing, Imperial College, 180 Queens Gate, London SW7 2BZ. Telephone: 01-589 5111 ext 2732 or 2740

London (Imperial College)

References

1. J.Kramer, J.Magee, M.Sloman and A.Lister, 'CONIC: An Integrated Approach to Distributed Computer Control Systems', *IEE Proceedings*, Vol. 130, Pt. E, No. 1, pp. 1-10, January 1983.
2. J.Magee and J.Kramer, 'Dynamic System Configuration for Distributed Real-time Systems', *IFAC/IFIP Workshop on Real-time Programming*, March 1983.
3. M.Sloman, J.Kramer, J.Magee and K.Twidle, 'A Flexible Communication System for Distributed Computer Control', *IFAC DCCS Workshop*, May 1983.

PROFESSOR G D CAIN and R C S MORLING
THE POLYTECHNIC OF CENTRAL LONDON

LOCAL AREA DATA NETWORK FOR INSTRUMENTATION APPLICATIONS
April 83 - March 85

MININET is a local area network aimed at real-time scientific and process control applications. The network employs packet switching technology to achieve highly reliable interconnection of a heterogeneous assortment of instrumentation devices as well as mini and microcomputer hosts.

This network utilises short fixed-length packets (i.e. a "word switching" approach) to achieve ultra-high transparency to user devices which generally are not network conscious. Its high-speed word handling capacity provides the short trans-network response times for data or control bursts essential for meeting the real-time deadlines of on-line instrumentation. MININET has a flexible modular structure of arbitrary topology, composed of two main components (nodes which are either "Stations" or "Exchanges") which can be arranged in nearly any convenient layout. A multiplicity of different links can be supported, with one MININET cheerfully accommodating slow-speed links (like modems) and also high-speed links (such as fibre optics).

In OSI Reference Model terms, MININET is a true network since communicating nodes do not have to be connected to a common data link, but may transfer information via one or more intermediate nodes. MININET is unique in the degree of Network Layer ("Level 3") services which are being built in as inherent features. Users can construct their own end-to-end communication upon MININET without the need for host computers. Operational bottlenecks, such as *low transaction speeds*, that plague most other *Level 2* local area "networks" are avoided by provision of very high-speed special-purpose processors(the nodes) developed for MININET. The latest version of the Station (now under test) meets its specification of one megapacket per second internal processing speeds!

Packets 32 bits in length are interchanged between devices, transducers or computers attached to up to 64 nodes spread over an area typically of 2 km expanse. Each Station can accommodate up to 64 user devices (attached via the IEEE 488 interface or an intermediate interface known as DIM), so a sizeable user community can be served with easily-reconfigured "virtual connections".

MININET is being jointly developed by the Signal Processing Group at the Polytechnic of Central London (PCL) and the Automatics Institute at Bologna University. Support has come from a succession of grants awarded by SERC, the Italian CNR, NATO and EEC. Plans for the coming year call for an increasing effort to be directed to pre-product development activities in consultation with industry, in parallel with the on-going research aspects.

Earlier demonstrations have centered on the "computer room" type of resource sharing environment that has been continually served by a low-speed prototype Station at Bologna University since February 1980. There several small computers have been flexibly connected into a pool of slow peripherals such as line printers and paper tape readers. Incorporation of the first full-performance Station during 1983 into an environment at PCL emphasising support of real-time speech data collection with remote digital signal processing is expected to provide useful stress testing experience for MININET's demanding high-speed instrumentation role.

Future research activities are being concentrated on efficient incorporation of popular Level 2 multipoint data communication media, such as rings and buses, as special cases of MININET channels. Utilisation of multiple Cambridge Rings appears to be a particularly fruitful possibility. Near-term objectives include development of an interface to a Cambridge Ring and an IEEE 488 network port.

PROFESSOR Y PAKER
POLYTECHNIC OF CENTRAL LONDON

COMPUTER AIDED MULTI-MICROPROCESSOR SYSTEMS MODELLING, SIMULATION AND EVALUATION
April 79 - March 81

Background

Microprocessors and microcomputers are being coupled together in increasingly large numbers, reaching several hundred or more, in a tightly or loosely coupled manner as distributed computing structures which include complex interconnection mechanisms and interfaces to link these to an application. Superimposed on this hardware structure, software is written to provide the communication protocols, synchronisation between sequential processes, application programs and so on. Considering the complexity of such systems, however, currently there are no adequate methods to understand and evaluate many of the basic architectural and software options that exist when a distributed computer system is designed to satisfy the requirements of a particular application environment.

Achievements

A software design package called MICROSS has been written which is capable of giving a designer the ability, by means of an interactive terminal, to model his system using standard, or user defined building blocks and their interconnections. Once a model is defined, then a number of simulation runs can be performed to derive principal performance figures such as delay times and throughput. MICROSS is currently on a DEC 10 computer accessed by means of a Tektronix 4014 terminal.

References

1. Y.Paker, H.English and M.Bozyigit, 'NPL Multicomputer Ring Modelling and Simulation', in *IFIP WG 6.4 Workshop on Ring Technology*, Canterbury, September, 1983, ed. I.N.Dallas and B.Spratt, North-Holland Publishing Company, 1983.
2. M.Bozygit, H.English and Y.Paker, 'MICROSS: Graphics Aided Simulation of Distributed Computer Systems, in *Synchronization, Control and Communication in Distributed Computer Systems*, ed. Y.Paker and J.P.Verjus, Academic Press, 1983.
3. Y.Paker and M.Bozygit, 'Computer-Aided Multi-Microporcessor System Modelling, Simulation and Performance Evaluation', Project Report 1, PCL September 1979.
4. M.Bozygit and Y.Paker, 'Computer-Aided Multi-Microprocessor System Modelling, Simulation and Performance Evaluation - MICROSS', Project Report 2, PCL December 1979.
5. M.Bozygit, 'A Dense Variable Topology Multi-Computer System: Specification and Performance Analysis', PhD Thesis, PCL May 1979.

London (Queen Mary College)

PROFESSOR G COULOURIS, R BORNAT and S ABRAMSKY
QUEEN MARY COLLEGE

DISTRIBUTED SYSTEM REQUIREMENTS FOR EFFECTIVE MAN MACHINE INTERACTION
and
DEVELOPMENT OF A DISTRIBUTED PASCAL-M SYSTEM
Nov 81 - Oct 83

Background and Nature of Research

Powerful personal computers (eg ICL PERQ, Stanford SUN, etc.) have become affordable, and can be connected to form a computer network. If such networks are to become generally useful, they must offer more than just communication facilities. For example, the user sitting at the computer may wish to access shared information resources, use shared services such as printers, communicate with other users, and so on. As the tasks performed by computers, and networks of computers, become more complicated traditional styles of human/computer interaction become inadequate. New forms of information display and highly interactive control must therefore be developed to permit the most effective and natural human/computer interfaces.

This requires software which can manage resources, control access to services and handle the necessary communications both between computers in a network and between human and computer. Such software has tasks similar to, but more complicated than, the operating system software of a conventional time-shared machine: the complexity of its construction puts it at the limit of current programming technology. Novel programming methods therefore need to be developed together with a methodology for the design of internally consistent and practically effective user interfaces.

The software of a network can be viewed as a system of communicating but independent processes. This view of software simplifies the design of the operating systems of computer networks, and also those portions of the software concerned with highly interactive user interfaces. Pascal-m is a language which permits the construction of systems of communicating processes with these three properties. It is an extension of Pascal, intended to aid the design and construction of programs as collections of communicating processes. The main feature of the language is synchronised (unbuffered) typed communication over channels called mailboxes: mailboxes have identifications which may themselves be communicated in messages to alter the pattern of process interconnection. Non-determinism is provided primarily by the use of guarded commands. The current Pascal-m implementation is being used for programming user-oriented applications which have been the subject of design studies carried out by the EMMI project.

Research Goals

1. To develop methods for the construction of software for networks of personal computers.
2. To demonstrate these by the construction of experimental systems.
3. To develop a methodology of user interface design.
4. To construct experimental user interfaces as systems of communicating processes.

Achievements

The language has been implemented on a personal computer based on an 8086 micro-processor (using a cross-compiler running on a PDP-11 UNIX). It has been used to construct several experimental systems, including a demonstration of a model user interface based on a window display which was used to test the use of colour, sound and the visual display of multiple work-contexts in human/computer interaction. Another experimental system contained a hierarchical filing system which it is intended will eventually form the basis of a stand-alone Pascal-m development system. A message-protocol has been developed and implemented, which provides reliable communication between computers in a network and between processors within a personal computer. The demonstration systems have been developed into a complete applications environment (called QDOS), which includes a file system similar to that of UNIX, an input/output system supporting teletype, screen editing and menu styles of interaction, and supports several activities concurrently using multiple screen windows.

The EMMI project has been concerned with the design of powerful user-oriented integrated systems, with particular reference to office systems. The collaboration of the two projects has been intended to provide, via Pascal-m, a suitable programming system, and to use it to program user-oriented applications systems, in a system demonstrating a user interface designed around a clearly defined user's model has recently been implemented.

Work in Hand

During the next three months, up to the end of the project, we plan complete translation of the compilation system to the QDOS environment, and to demonstrate a distributed implementation of Pascal-m using the Cambridge Ring. This will complete the current project. Several projects to continue and extend are being planned.

Applicability

Obvious areas of the application include office automation, educational use of microprocessors, software for multi-processor workstations. The concepts and techniques developed in the research are directly useful to industry in constructing these types of systems. Industrial concerns which have already expressed positive interest and offered financial support or equipment for the work include ICL, Texas Instruments, GEC and Hewlett Packard.

Further Information

Further information can be obtained from any of the investigators.

References

(These are internal publications available from QMC)

1. S.Abramsky and R.Bornat, 'Pascal-m, a Language for Distributed Systems', Report No. 245
2. S.Cook, 'ANY: A New Primitive Type for Pascal-m', Report No. 299.
3. S.Abramsky, 'A Network Implementation for Remote Communication in Pascal-m', Report No. 275.

London (Queen Mary College)

4. S. Abramsky, 'A Pascal-m Filing System', Report No. 298.
5. R.Bornat, 'Formal Description of Pascal-m', Report No. 274.
6. S.Cook, 'Research into Distributed Information Systems at Queen Mary College', Report No. 296.
7. R.Bornat, 'The 30-year Truthfulness Problem', Report No. 304.
8. S.Abramsky, R.Bornat and G.Coulouris, 'Research into Distributed Software Systems and Applications at QMC', Report No. 314.
9. S.Cook and S.Abramsky, 'Proceedings of the 2nd International Workshop on Office Information Systems', Couvent Royal de St. Maximin, North-Holland, Amsterdam, 1982.
10. S.J.Cook, 'Playing Cards on the PERQ, an Algorithm for Overlapping Rectangles', to be published in *Software Practice and Experience*.
11. S.J.Cook, 'Pascal-m Primer', QMC Report No. 335, June 1983.
12. S.J.Cook, 'The Distributed Implementation of Pascal-m', QMC Report No. 339, June 1983.
13. R.Bornat, 'Programming Styles in Pascal-m', QMC Lab Report.
14. K.M.Clarke, 'Another Pascal-m Filing System', QMC Report No. 329.
15. R.M.Newman, 'User Interface Design for a Cut and Paste Editor', new QMC Lab Report.

PROFESSOR J K ILIFFE and P GRIFFITHS
QUEEN MARY COLLEGE

DESIGN STUDIES FOR ACTIVE MEMORY ARRAYS
Jan 83 - Jan 84

Background

The purpose of this research is to investigate the design and application of a new computer offering very high reliability, flexibility, and potentially high speed. It can be viewed as an extension and integration of ideas found in current machines such as ICL DAP, Intel iAPX 432, and ICL PERQ. However it seeks to avoid the systematic shortcomings of such machines.

Programme of Work

The SERC support is for one RA for 12 months and up to 9000 pounds in costs for a feasibility study. The effective starting date was March 1983. The intention is to produce detailed logic design and costings for a prototype machine.

Initial design of a planar arithmetic unit, bus structure, memory control and instruction coding have been completed. The design of a scalar arithmetic unit is now in hand. The initial design, including I/O and interrupt system, should be complete by November 83.

Possible construction methods are being investigated as design progresses, with a view to evaluating components at board level in the final stage of the study.

As the result of work carried out so far some useful economies have been made in the design. These will be reported in due course.

Support Activity

The "PN machine" has been interpreted at instruction level on the PDP-11 for some years. A simple operating system and programming language have been developed. The interpreter has been rewritten to use the Motorola MC68000 processor, for which basic support software has been written. Included in the interpreter are counters to assist in the evaluation of different logic design options.

A Ph.D student (M Burton) has undertaken redesign of the operating system to take advantage of the parallel arithmetic and logic functions and to refine the program protection mechanisms.

Application

The intended application areas include image analysis and transformation, high integrity systems, language-oriented interpretive systems, support for database access and for program management, eg. in compiling, debugging, error control. Separate funding will be sought for demonstrations in one or more of these areas.

Further Information

PN System: Users Manual, available from Department of Computer Science, Queen Mary College.

I PAGE
QUEEN MARY COLLEGE

DISARRAY - AN EVALUATION OF AN ARRAY PROCESSOR FOR BITMAP DISPLAYS
March 82 - Feb 85

Background

Disarray is a 256 processor machine for the execution of the RasterOp graphics primitive at high speed. It is constructed out of conventional (Schottky) TTL components but has an extremely regular architecture well suited to a later implementation in VLSI.

The basic architecture of the machine is similar to the ICL DAP in that it consists of a square array of 16 x 16 single-bit processing elements (PE's) each with their own local memory. The PEs have a four-nearest-neighbour connective with additional row-wise and column-wise input buses and column-wise (open collector) output bus. The processor array operates as an SIMD machine with a 16 x 16 square word which corresponds to a similar sized square patch of pixels. Such a square word is ideal for the RasterOp function and its size allows a high computational bandwidth for this (and other) graphics algorithms.

The major differences between DisArray and the DAP type of architecture can be listed as follows:

1. The PEs have a significantly reduced complexity (by a factor of about 50%) which trades its arithmetic computational speed (not needed for RasterOp) for simplicity.
2. The PE architecture is optimised for the fast movement of pixel data using the nearest neighbour connections, (currently over 7 Giga Bits/sec).
3. The processor array has an integral high-bandwidth output path in the form of a video shift register which passes through each PE. The corresponding serial input channel (for video capture) has not been implemented in the prototype.
4. The processor array was designed to be intimately integrated with a 16 bit micro-programmable host machine, the two forming an advanced (fifth generation) single-user workstation.

Progress

As of August 1983 the array processor is functioning with only a few minor problems outstanding. Pixel data can be shifted at over 7 Giga Bits/sec, memory-register computations can be performed at around 500 Mbits/sec, and some test programs have been written which produce simple animated displays on the integral 512 x 512 screen.

The generation of the instruction stream for the array processor is currently being carried out by the PDP11 host itself, which severely limits the throughput of the system. The real DisArray controller has been constructed and is in the process of being integrated into the system. The controller is a micro-programmed AMD-29116 which runs at about 7MIPs and will allow DisArray to run at its full speed. It is hoped to have the complete system running by December 1983.

Conclusions

Despite delays, DisArray is still a novel and significant advance in the area of high performance display systems. By integrating DisArray into a more powerful host an advanced fifth-generation single user workstation is obtained. Further, by improving its arithmetic capabilities the array would also be able to support a large amount of numerical vector/array processing for graphics, modelling, simulation etc. Coupled with the high graphics speed that DisArray gives would make the resulting machine a really exciting research vehicle. Without even going to specially designed chips it is possible for the next version of DisArray to offer about an order of magnitude speed-up over the current hardware, 4 Mbytes of memory and also over 300 MIPs of vector processing (add, subtract, logical and shift operations on 16 bit quantities) together with some content addressable memory features such as 32 simultaneous 16 bit comparisons or 512 1-bit comparisons.

London (University College)

P L HIGGINSON
UNIVERSITY COLLEGE, LONDON

A STUDY OF PROTOCOL REQUIREMENTS FOR MULTI-MEDIA MESSAGES AND INTERCOMMUNICATING OFFICE MACHINES
Jan 83 - June 85

This project aims to study the protocol problems and requirements for multi-media message exchange.

X25 Packet Switching Networks are now well established and there are accepted UK protocols (even if not yet international ones) for Transport Service, file and job transfer. There is an interim JNT mail protocol which is a combination of the well established Arpanet message structure format and the UK File Transfer Protocol. Whilst the Transfer part can handle any data format, the structure is restricted to ASCII text. This is ideal for interim use, but effectively prevents the transmission of other types of data such as graphics, voice, video and facsimile as a part of messages. These defects were known to the definers of the protocol, but were necessary for compatibility with existing procedures and appropriate in an interim protocol.

There is a new set of Teletex Recommendations (ie standards) from CCITT, which are likely to shape future work in this field. In many areas (for example modems) international standards follow from the developments of acceptable prototypes and their use in real situations. Only in data communication do we see "seat-of-the-pants" design resulting in international standards without any significant implementations being done. For example, X25 and X3, X28 and X29 were produced in this way; they are still being changed at an alarming rate and have serious fundamental weaknesses which there is no hope of correcting.

The Teletex service and protocol, whilst stated to be memory to memory, is oriented towards stand-alone, dedicated machines. The present specification caters only for text transmission although there are plans to adapt it to graphics working. Networks and Teletex customers are planned, and are likely to be an experimental phase during the course of the project.

The CCITT Teletex Recommendation is structured as a number of documents which describe the requirements and general facilities, the basic service, the presentation functions, the end-to-end or session functions and the transport service. As it is one of the first protocols to be designed using the ISO/OSI model; although the model is not yet finalised, there is a clear convergence between it and the Teletex standards. Revisions to Teletex are likely to bring it even closer to the model, and for this reason it is an important protocol to study.

The transport service for Teletex is generally acknowledged to be extremely important and quite adequate, since it is based on the ECMA draft standard (on which much work has been done). The higher levels are inadequate for the long term needs of information transfer, and much work remains to be done. The GILT project is also investigating some of these areas. We have been asked to participate, but so far have lacked the effort to do so.

While early work in facsimile has followed a different direction from Teletex, the future Datafax work should be fully compatible with Teletex. Early Video Gateway work has also been on a different basis, however, the current CCITT thinking is also along ISO/Teletex lines. We have joint activities with BT Martlesham in this area.

This project is intended to show that certain communications and methods are feasible to small and medium sized office systems.

In the initial period the software work has concentrated on completing and enhancing a distributed Teletex implementation which was started under another grant. We have demonstrated ability to to send Teletex messages across networks of the Universe type as well as X25 ones and plan further tests to ensure our compatibility with industry standards. We have joined the BT Industry group on Teletex. A Codata 68000 has been obtained to fulfil the small workstation role in the original proposal, and a special high speed interface for connection to ISDN networks is being built. Some work has also been done (in co-operation with the UCL Universe group) on improving the facilities to boot small distributed workstations.

Further Information

For further information contact P L Higginson, Department of Computer Science, Tel 01-387-7050.

London (University College)

PROFESSOR P T KIRSTEIN
UNIVERSITY COLLEGE LONDON

COMMUNICATION PROTOCOLS IN THE CONTEXT OF X25 COMMUNICATIONS PROJECT
Oct 78 - Feb 83

Aims

The aim of the project is to investigate the features and performance of the PTT agreed X25 protocol and of the High Level protocols needed in a computer network consisting of both high speed Local Area Network (LAN) components and gateways to Wide Area Networks (WAN). The correctness, ease of implementation, adequacy of function, independence of level, level structure, dependence on sub-net properties and protocol performance are all being studied.

Experimental Environment

The experimental environment is three Cambridge Rings with a gateway to the X25 PSS and SERCNET X25 networks. For comparison purposes there is also a gateway to the DARPA SATNET and UNIVERSE satellite Neworks. A number of minicomputers on a ring can provide protocol translation, performance and availablity measurements, and gateway function. About a dozen PDP 11 or LSI 11 computers participate in a distributed way in these functions.

Changes in Direction

Over the last year the specific needs for good facilities for network measurement and control have been recognised. Considerable effort has gone into the provision of such facilities and introducing similar techniques with a follow-on project. Since the start of the project, the standards for document service over such networks have emerged. The protocols of concern have broadened to include Videotex, Teletex, Computer Based Message Systems (CBMS) and Facsimile over such networks. There has been particularly heavy emphasis on CBMS.

Recent Progress

The relevant publications written during the last 12 months are listed in the references; there are numerous additional internal documents.

The project officially terminated in February 1983, so that this report really represents only about 5 months work. Most of our activity was, therefore, devoted to finishing off the work in preparation for new projects. Many of the isolated strands have, therefore, been taken up in new projects.

The work in interconnection architectures was largely completed. We were able to show how to provide the different protocol levels in a completely distributed way, so that the same high level protocols could operate over different low level ones; alternately different high level ones could operate over the same low level protocol engines. Our work with the same file, terminal and mail protocols over the DARPA internet and the X25 networks is discussed in [1] and [5]. This work has been demonstrated by tests with Teletex; here implementations distributed around the UCL Rings, use the same X25 gateways as the other applications.

Naming and addressing have been a problem at several levels of protocol. Our work at the transport and network access level are discussed in [2]. We have become much more active in Mail across different networks. Here again protocol differences have caused us many problems, some of which are discussed in [6]. This work is continuing in a new project under Martin.

The monitoring work has continued, partly with help from this project and partly in the context of project UNIVERSE. One publication in this area is (8). We have now achieved a good ability to monitor the status of all the network connected to UCL (3 Rings, SERCNET, PSS, SATNET/DARPA CATENET, UNIVERSE), and have a fairly detailed understanding of how such systems should be constructed. Again the work is continuing, but now entirely under UNIVERSE auspices.

As part of the improvement of network reliability, there has been an investigation of automatic bootstrap loading access the Cambridge Ring. While this work has now been largely successful [4], only some has been achieved under the auspices of the present project. It has been continued in a project under Higginson, which started early in 1983.

The theoretical analysis has concentrated largely on UNIVERSE applications [7]. It was clearly that this work would pay highest dividends over the current year. Again it has been continued under UNIVERSE auspices, but will be an independant project from October 1983 under Leung and Kirstein.

Industrial Involvement

The involvement in the Teletex work was with BT and Logica. From their part it was under UNIVERSE auspices; from ours it was partly from this project, and partly under other funding. In any case, the collaboration has been strong. ICL has had a staff member on the group in monitoring activities throughout the year. CASE Ltd has been sufficiently interested in the monitoring work to fund a studentship in this area. GEC is collaborating also with this work - through their involvement is again under UNIVERSE auspices. The CASE studentship from BT has continued the collaboration on distributed Videotex.

Further Information

Further Information can be obtained from Professor P T Kirstein.

References

1. R.Cole et al, 'A flexible architecture for protocol studies in a multi-network environment', *IFIP83, Paris*, pp 401-406, 1983.
2. R.Cole et al, 'Naming, addressing and routing in a multi network environment', Data *Communications 12, 6* pp 175-187, 1983.
3. R.Cole et al, 'An internetworking architecture for STELLA and UNIVERSE,', *Proc. Conf. Satellite and Computer Communication*, North Holland, pp 913-918, 1983.
4. J.Crowcroft and R.Hall 'Distributed Approach to loading in a network environment', *International Workshop on Ring Technology*, Canterbury, pp 111-122, 1983.
5. P.L. Higginson and R.Cole, 'Issues in Interconnecting local and wide area networks', *Proc. Business Telecom Conf, ONLINE*, pp 231-241, 1983.
6. S.Kille 'The interconnection of multiple international mail systems using different addressing strategies', *Proc, Symp. Computer Based Mail Systems*, Versailles, 1982.
7. S.A.Sorensen, 'Simulation studies of a UNIVERSE type network, *Proc. Conf. Satellite and Computer Communication*, North Holland, pp 343-352, 1983.
8. S.R.Wilbur, 'Aspects of Distributed Systems ManagementΞq, *International Workshop on ring Technology*, Canterbury, pp 111-122, 1983.

London (University College)

N MARTIN
UNIVERSITY COLLEGE, LONDON

MESSAGE SERVICES AND DIRECTORY DEVELOPMENT
July 83 - June 85

Aims

There have been many developments in the area of Message Transport Protocols. The aim of this new project is to investigate some of the more significant ones.

The current system at UCL is based on the JNT Mail Protocol, which uses a general purpose file transfer mechanism (NIFTP) as the mechanism for transporting the messages. It is proposed to evaluate other message formats and message transfer mechanisms. Although the emphasis is intended to be on the latter, the problems of message representation are closely tied in with the transfer of problems and will be considered in that context, as opposed to the difficulties in the area of user presentation. The research is expected to comprise a mix of theoretical and practical evaluations of the various protocols.

1. The CCITT standard transport protocol Teletex will be investigated as a medium for transporting messages. This work is dependent on the provision of the Teletex protocol at UCL: the Teletex implementation is being completed under an ongoing activity. It is proposed that JNT Mail will be extended to run on Teletex as an initial experiment in this area.
2. This will later be extended to an investigation of the GILT protocols, which aim to be more sophisticated Message Protocols following the Teletex model.
3. It is proposed to continue the investigations of the Internet Message protocols based on the new message format which has recently been adopted. In particular the Protocol SMTP (Simple Message Transfer Protocol) which will be implemented and evaluated.
4. The problems of interconnection by use of telephone networks will be examined. Two protocols will be considered in parallel. The first is the CSNET/Phonenet system developed at the University of Delaware, which fits in very naturally with the mail software currently running at UCL.
5. The second utilises UUCP and is currently used by Usenet in the USA along with a limited number of European sites engaged in rather ad-hoc communications. It is considered to be a significant pragmatic protocol for Unix sites.
6. The protocols relating to the BT GOLD/Dialcom system will be examined, particularly in relation to compatibility with other systems. The Telemail protocols may also be considered.
7. The work of various standardisation bodies in this area is considered to be very important in this context. The work will be described in terms of the models and standards of ISO, CCITT, IFIP, and NBS in particular. This work will be continuous over the period of the project.

An important section of the work will be to determine the problems which result from interconnecting the systems being investigated. The difficulties associated with performing protocol conversions, and defining the services which can and cannot be mapped will be assessed. It is seen as important, to both the end user and each individual system, that it be possible to view the set of interconnected mailsystems in a homogeneous manner. This means that each system should in some way view the others as a part of itself. Some of the difficulties of this have already been analysed at UCL. It is hoped to extend this work. In particular work will be undertaken to build sophisticated address and format translation facilities.

A new area which will be considered is that of access control and accounting. The difficulties of protecting and charging for use of an expensive or restricted resource have not yet been analysed carefully in the context of electronic mail. It is clear that these facilities must be available at the mail level. The protocol features necessary to achieve this will be evaluated. Experimental implementations are planned.

It cannot be overemphasised that these problems cannot be completely resolved by purely paper studies: only by implementation exercises, backed up by practical evaluation, can the complete problem be studied.

Further Information

For further information contact N Martin, Department of Computer Science. Tel 01-387-7050 Ext 818.

DR P E OSMON
WESTFIELD COLLEGE

IMPLEMENTATION OF A HIGH LEVEL DATAFLOW PROGRAMMING LANGUAGE
July 78 - Sept 81

Background

The project has been concerned with the fundamental principles underlying dataflow. The investigators have developed a set of definitions and primitive concepts, and have designed a higher order functional language CAJOLE, consistent with their model.

Research Goals

To elaborate the language definition and model; to develop compilers for the CAJOLE language; including one for the Manchester dataflow machine to assess the results.

Achievements

During the course of the project, several compilers for CAJOLE have been produced, together with a supporting program development system, employing a graphics terminal (which proved attractive to its users). The Manchester machine emulator was ported to the Westfield site and the compiler for the Manchester machine was successfully used to compile CAJOLE programs to run on the emulator. A considerable amount of work on the formal specification of CAJOLE and dataflow programs was carried out.

PROFESSOR D J EVANS, DR I A NEWMAN and DR M C WOODWARD
LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY

A RESEARCH VEHICLE FOR INVESTIGATING THE USE OF CLOSELY COUPLED DISTRIBUTED SYSTEMS
Dec 79 - May 83

Background and Nature of Research

The project arose originally out of the conjecture that a general purpose computer system could be configured in a cost-effective way from a collection of interconnected minicomputers. It was taken as axiomatic that some application areas exist where most tasks are suitable for processing within the resources offered by a single minicomputer with a few tasks requiring greater power (faster speed) than could be offered by a single processor. A previous research contract investigated the feasibility of linking dissimilar minicomputers together while providing a uniform command language interface. It also demonstrated that a twin processor system where the processors shared part of the memory could be used to achieve execution speeds close to twice those of a single processor for particular algorithms for problem solution. The present research seeks to extend the investigation to demonstrate that it is feasible to link together both more and more sophisticated machines cheaply, without affecting their capability to act as uniprocessor systems and yet still yielding a high percentage of the possible speed up for the greater number of processors. Fundamental assumptions were that the work should be based on commercially available hardware; that a minimum of changes should be made to the operating system and utilities; that special software should be avoided wherever possible.

Research Goals

The main goal was to establish a working multiprocessor system with support for parallel program development, and to make it available to the DCS community as soon as possible. Subsidiary goals were: to investigate alternative methods of managing the resources that were shared between the processors (specifically shared memory); to repeat the demonstrations of 'reliable' code which have been given on the earlier system; and to investigate the provision of algorithms for various tasks on the multiprocessor system to ensure that the 'promise' of such a system could in fact be realised in practice.

Achievements

Despite problems in obtaining the hardware and suitable research staff, a working system was made available to the DCS community within nine months of the acceptance of the hardware. A seminar was held at Loughborough and several external users have subsequently used the system to develop and test algorithms. A large number of algorithms have been written in the Department to exploit the parallelism of the system. Many of these have already been published as research papers in journals or at conferences, and they provide a convincing demonstration of the power and flexibility of the multiprocessor system. A new method of coordinating access to shared resources is also a worthwhile product of the research.

Loughborough

Work in Hand

The work on different algorithms to exploit parallelism is being continued by several research students as well as research assistants. Improved algorithms for managing the shared memory and multiprogramming multiprocessor and uniprocessor tasks are being developed and an integrated management scheme for the shared disc is under investigation.

Applicability

The system demonstrates the viability of assembling and programming multiprocessor systems for general purpose use. Texas Instruments are the only firm to whom the results are directly useful and they are showing some interest.

Further Information

Contact the investigators.

References

'A Guide to Using the NEPTUNE Parallel Processing System', Internal Publication.

Many other publications. A list can be obtained from the investigators.

PROFESSOR D J EVANS and DR I A NEWMAN
LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY

AN INVESTIGATION OF THE RELATIONSHIP BETWEEN ALGORITHM STRUCTURE AND PARALLEL ARCHITECTURES

July 79 - Dec 82

Background and Nature of Research

Parallel computers, that is computers with more than one processing unit, are the only solution to enormous demands for computing power from extremely large problems or tasks with severe real-time response constraints. A further impetus to the viability of parallel computing comes from noting that the advent of large scale mass production of computing units could offer a more cost effective, more flexible and potentially more reliable provision of computing resources through the coupling together of a number of basic units.

However, the provision of computing power through a number of such units does not automatically mean that all the units can be effectively used. The workload has to be shared out evenly. This particularly applies to using more than one computing unit on the same task: this is the usual interpretation of parallelism.

Furthermore there are overheads which are associated with loading the program into the various units, the communication of results between the units and finally the possible synchronisation of the states of subtasks on different units.

Research Goals

The existence of several accessible parallel computers has oriented the research to both practical and theoretical aims which include the following:

1. To investigate the design and implementation of algorithms to solve a number of large scale numerical problems using machines with different parallel architectures.
2. To provide a general theoretical framework from which the efficiency of different algorithmic structures on various parallel computers could be predicted.

Achievements

1. Practical Level: The investigators have developed the best (to date) available parallel solutions for the symmetric eigenvalue problem for all types of parallel computers. Dynamically the investigators' procedure chooses the optimum strategy for the machine and problem characteristics [1]. A sparse representation of banded matrices (as arise in most structural analysis programs) has been developed for use on array processors [2]. Other work of interest has been published [see 3].
2. Theoretical Level: The analysis of a problem for parallelism examines the various options for problem decomposition and characterises each option in terms of its potential processor utilisation, and its communication and synchronisation costs. From this the most effective solution for a given system can be chosen. This analysis was originated by the Loughborough group [4] and has developed further in recent work.

Loughborough

Applicability

The work on sparse representation referred to above can be embedded into a well known iterative method for the solution of linear systems (Conjugate Gradient Method), and eigenvalue problems (Lanczos Algorithm).

References

1. R.H.Barlow, D.J.Evans and J.Shanehchi, 'Comparative Study of the Exploitation of Different Levels of Parallelism on Different Parallel Architectures', *Proceedings of the 1982 International Conference on Parallel Processing (IEEE Computer Society Press)*, pp 34-40.
2. R.H.Barlow, D.J.Evans and J.Shanehchi, 'Sparse Matrix Vector Multiplication on the ICL-DAP', Internal Report No. 161.
3. R.H.Barlow, D.J.Evans and J.Shanehichi, 'Performance Analysis of Algorithms on Asynchronous Parallel Processors', *Computer Physics Communication*, Vol. 26, pp 233-236 (1982).
4. R.H.Barlow and D.J.Evans, 'Analysis of the Performance of a Dual Minicomputer Parallel Computer System', *Proceedings Eurocomp 1978*, Online Conferences, Uxbridge, pp 259-276.

DR J R GURD and DR I WATSON
MANCHESTER UNIVERSITY

A MULTILAYERED DATAFLOW COMPUTER SYSTEM
Oct 81 - Sept 85

Background and Nature of Research

This project is for the development of a parallel computing system based on the dataflow model of computation. This model yields irregular concurrency at the instruction level. It allows a wider range of applications than the more rigid vector and array processors.

Research Goals

The primary short-term goals are to complete implementation of the prototype single-ring system now in operation; to conduct further analysis of its performance; and to tune the existing hardware for optimum technology-bound performance. The long term goals centre around construction and tests of a full multi-ring system; investigation of storage hierarchy management in both single and multiple rings; and the techniques for parallel dataflow algorithm design.

Achievements

By summer 1983, the single ring prototype was about 90% complete, and had achieved speeds of up to 2 MIPS. Preliminary performance evaluation completed, with encouraging results - without hardware tuning, performance improvement through concurrency is almost linear for up to 10 processing elements. A maximum of 15 processing elements have been in operation. A new Matching Store using dynamic RAM has been designed and is about to be commissioned. A multi-ring simulator based on M68000 hardware has been constructed. A new front-end system has been installed.

Work in Hand

Hardware upgrade and tuning now in progress will be followed by further evaluation exercises. The experimental system is now supported by a large front-end computer providing fuller operation and development facilities. Studies in store utilisation are underway, as is work on parallel algorithm transforms. As well as the investigators' team, the prototype is being employed by several external users to test a variety of applications - providing very useful feed-back on design and performance. Participation by external users in this way will be further encouraged, supported by the new front-end facility and a DCS-supported User Liason Officer.

Applicability

The prototype system is immediately useful for testing dataflow codes on real hardware - this may well be valuable to industry in assessing the commercial viability of dataflow products. These products themselves could be applicable to areas with irregular problem parallelism: these include numerical applications such as simulation and automated design, and, possibly, to non-numerical areas such as data processing and artificial intelligence. One computer manufacturer is already testing CAD algorithms on the dataflow hardware.

Manchester

Further Information

For further information please contact the following people: Dr J Gurd, Dr I Watson, B Saunders or Mrs V Bush.

References

1. I.Watson and J.R.Gurd, 'A Practical Dataflow Computer', *IEEE Computer*, Vol.15, No. 2, p 51, February 1982.
2. J.G.D. da Silva and I.Watson, 'A Pseudo-Associative Store with Hardware Hashing', *Proceedings of the IEE*, Vol.130E, No. 1, January 1983.
3. J.R.Gurd and I.Watson, 'Preliminary Evaluation of a Prototype Dataflow Computer', *Proceedings of the 9th World Computer Congress, IFIP '83, North Holland, September 1983*.

DR R N IBBETT, DR D A EDWARDS and DR C J THEAKER
THE UNIVERSITY OF MANCHESTER

A HIGH PERFORMANCE OPTICAL FIBRE BASED LOCAL AREA COMPUTER NETWORK
June 83 - Oct 84

Background and Nature of Research

Local area networks have generally been developed in response to a need for resource sharing among a group of computer users concentrated in a small geographical area, typically within a single building. At the University of Manchester there are many groups of users concentrated in a number of buildings scattered about a large campus. Many have computing facilities of their own but virtually all require access at some time to the central facilities provided by UMRCC (the University of Manchester Regional Computing Centre). UMRCC, furthermore, is housed in the same building as the Department of Computer Science, which has traditionally been involved in the design and implementation of the hardware and software of large computer systems and which is currently involved in the implementation of a large multi-computer system, MU6. Centrenet has therefore been designed as a system capable of satisfying the requirements, not only of a closely knit multi-computer system such as MU6, but also of a scattered community of users who wish to transfer files between their own machines and the central site, to gain terminal access to a variety of systems, and to share a variety of hardware and software resources.

Research Goals

The principal aim of the project is the implementation of a high performance networking system, but this will entail research in a number of areas. These include the technology of switching and transmission mechanisms, network and internetworking architectures, network management software, network performance measurements and protocols for operating system interaction and for session level interaction.

Achievements

Some hardware designed and implemented as part of an earlier pilot project supported by Department funds is now operational.

Work in Hand

Software interfacing within MUSS to the hardware of the pilot project is under way. Some further pilot project hardware is being commissioned and some is being re-designed. Network management software is being written and partially tested on separate processing equipment.

Further Information

Contact Dr R N Ibbett, Department of Computer Science, University of Manchester.

DR C C KIRKHAM and DR J R GURD
UNIVERSITY OF MANCHESTER

DEVELOPMENT SOFTWARE FOR A PROTOTYPE DATAFLOW COMPUTER
March 81 - Feb 84

Background and Nature of Research

This project is to develop basic software techniques for compilation from high-level programming languages to parallel dataflow hardware.

Research Goals

The project intends to link the prototype dataflow machine at Manchester to the SERC network and provide a software distribution scheme for dataflow experimenters, with a set of software tools. The project intends also to survey the field of potential high-level dataflow languages, and to provide a dataflow compiler for the most suitable language - selected or designed.

Achievements

A portable dataflow simulator suite, including an assembler based development facility, has been produced and is available for distribution on tape, including documentation. This suite has also been successfully transferred to the new front-end system. A language survey has been completed [2]. A new single-assignment language SISAL [3], has been designed in collaboration with a major computer manufacturer and a National Laboratory in the USA.

Work in Hand

Final work is in progress on a compiler for SISAL. Dataflow code is expected to be produced within the next few months. There will then be some time to investigate optimisation.

Applicability

The tools now available are most useful to the dataflow language compiler writers working at lower level. The tools are useful to other experimenters in designing codes for simulation and hardware trial, including industrial workers. The SISAL compiler will have more general applicability for end-users.

Further Information

For further information contact Dr C C Kirkham, or Dr J R Gurd, Department of Computer Science.

References

1. J.R.Gurd, J.R.W.Glauert and C.C.Kirkham, 'Generation of Dataflow Graphical Object Code for the Lapse Programming Language', *Lecture Notes in Computer Science*, Vol. III, p 155, June 1981.
2. J.R.W.Glauert, 'High-level Languages for Dataflow Computers', *Pergamon-Infotech State of the Art Report on Programming Technology*, p 173, March 1982.
3. J.McGraw, S.Skedzielewski, S.Allan, D.Grit, R.Oldehoeft, J.Glauert, I.Dobes and P.Hohensee, 'SISAL: Streams and Iteration in a Single-Assignment Language', *Language Reference Manual*, Version 1.0, July 1983.

PROFESSOR D ASPINALL
UNIVERSITY OF MANCHESTER INSTITUTE OF SCIENCE AND TECHNOLOGY

THE USE OF MICROPROCESSORS IN INFORMATION SYSTEMS
April 82 - Sept 85

Background and Nature of Research

Many information processing tasks involve several concurrent processes. The low cost of microprocessor technology suggests that these may be active simultaneously on several processing elements or microcomputers, with a benefit in terms of cost and resilience over a time-shared mono-processor. This project has been concerned with the design and construction of a multi-microprocessor system; CYBA-M, which specifically uses low cost, commercially available microprocessors as the processing elements. In this system there are 16 processing elements which communicate through a Direct Shared Memory. This phase of the project is now complete and the CYBA-M is being used as a research vehicle in a variety of studies which are aimed at establishing the role of this type of computer system in industry and commerce.

Research Goals

The initial goals were the identification of firstly, the hardware and software design methodologies for this particular multi-microprocessor and secondly, the requirements of a software development environment for this type of system in general. These goals have now been achieved. The current long-term goal is to determine the architecture of the next generation of multi-processor systems using VLSI technology (Drs Edwards and Ritchings). The shorter term goals are concerned with the identification of the optimum machine architecture for a variety of application studies.

Achievements

The CYBA-M hardware and software development environment (EMU) are now operational and form a completely integrated multiprocessor development environment for the research community at UMIST. Links to remote users at British Telecom are also installed. The CYBA-M system can be used to demonstrate a variety of image processing algorithms and performance models of different multi-processor interconnection structures. The capabilities of a system such as CYBA-M for the control of complex industrial machinery has also been demonstrated for the case of a commercial print cutter. The print cutter was loaned for the duration of the study from ECM Engineering Design Ltd. of Bristol.

Work in Hand

Several members of the research team are involved with measuring the performance of various models, programming languages and image processing algorithms that have been implemented on CYBA-M. In addition a study is taking place to provide a better understanding of maintenance and fault finding techniques on a multi-processor. The connection with this work and the recent developments in IKBS technology are being examined and a prototype expert system has been developed that will assist in hardware fault finding activities on CYBA-M. The parallel implementation of an applicative programming language (SASL) is complete and is being used in this study. An investigation of algorithms and architectures for dynamic pattern matching applications in image and speech processing is also taking place.

Manchester (UMIST)

CYBA-M has also been used remotely by colleagues at Bristol University for the further investigation of design methodologies (Prof Dagless) and interconnection structures (Dr Barton) with particular reference to telecomms systems. In another study involving British Telecom (Mr Porteous), CYBA-M was used to model message passing in a small telephone exchange for performance measurements.

Several other research workers who are not part of this SERC funded team are currently using CYBA-M as a research vehicle. One group (Mrs Hughes and Dr Powell) is currently investigating the problems associated with implementing the language Pascal Plus on the machine. CYBA-M is also being used to evaluate ring message passing protocols in collaboration with the USAF. Finally, the Building Department at UMIST are using CYBA-M to model activities on a building site (Prof Pilcher).

Applicability

The results and knowledge that has been obtained can be applied to the design of any multi-processor system where each processing element is devoted to a specific task. Industry has been involved with the project through discussions and direct funding. Preliminary discussions have been held with INMOS in connection with simulations of new multi-processor systems while informal meetings have taken place with UKAEA and British Aerospace with regard to using multi-processors in their respective image processing applications. Industry has been responsible for funding the collaborative studies with the USAF, BT and ECM Enginering Ltd.

Further Information

The two key references to the project are 'Overview of Multi-Microprocessor Development Environment', (D.Aspinall and E.L.Dagless, in Microprocessors and Microsystems, Vol 3, No. 7 pp 301-305 1979) and 'EMU: a Multiprocessor Debugging Tool' (P.C. Burkimsher in Software and Microsystems, Vol 1, pp 41-47, 1982). Further information, including a complete CYBA-M documentation kit can be obtained from Prof D Aspinall.

J W HUGHES and M S POWELL
UNIVERSITY OF MANCHESTER INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTIPROCESSOR SOFTWARE ENGINEERING
Oct 80 -Sept 83

Background

The research stems from an investigation into the role of parallelism in the simplification of program designs. A language notation, DTL [1,2,4,9,16] has been developed for expressing a program design as a hierarchically structured concurrent network of sequential processes. Application of DTL to a major realistic problem is now the subject of a co-operative award made to the investigators in collaboration with BL systems Ltd [3]. Programs designed using DTL are capable of direct compilation or alternatively the DTL description may be used as a guide to implementation in a conventional programming language. The work on program implementation is investigating both the direct implementation of DTL programs and their indirect implementation in more conventional concurrent languages. In support of this work compilers and virtual machines for both languages are being implemented on loosely coupled and closely coupled multiprocessors as well as a uniprocessor system. A third aspect of this work is the formalisation of the semantics of DTL to facilitate reasoning about designs and verification of implementations.

Achievements to Date

Two versions of the DTL virtual machine are now operational and allow DTL programs to be executed in the UCSD development environment. The first of these employs conventional multiprocessing techniques and executes the tree structured code produced by the DTL compiler. The second version, is essentially an expression evaluator which copes with the concurrent features of DTL by supporting pipeline, parallel composition and cyclic composition operators. Programs are represented to the system in an abstract DTL syntax which resembles CCS [12]. This provides a very flexible implementation which can produce informative traces as an aid to reasoning about the behaviour of DTL programs. The DTL compiler has been modified to generate code in the required format.

Two Pascal based language implementations have been produced for the Cambridge Ring system. A distributed implementation of Concurrent Pascal [10] which allows unmodified Concurrent Pascal programs to be developed under the Solo operating system and executed with true concurrency on a variable number of processors connected to the ring. Process to processor allocation is performed using an interactive loader. Interprocess communication is achieved using the normal monitor mechanism and therefore no language extensions were required. A complementary Pascal Plus implementation for the ring [11] is based on an extended version of the Belfast portable Pascal Plus system [7] for the LSI-11. Again no language modifications were required and process to processor allocation is an interactive procedure.

Based on these experiments and the lack of a suitable development environment for Pascal Plus programs, an extended UCSD virtual machine was produced [6] with the intention of modifying the compiler correspondingly to allow Pascal Plus programs to be developed and executed in the UCSD environment. Similarly, in preparation for implementing Pascal Plus on CYBA-M, a UCSD virtual machine was implemented on CYBA-M and a means of controlling CYBA-M from a UCSD system running on the PDP-11/34 front end was provided. However since the last progress report, work described in the following section has indicated that a more radical redesign of the Pascal Plus virtual machine is necessary in order for it to be truly portable to distributed architectures. In particular the Belfast portable virtual machine makes a number of simplifying assumptions, which renders it insufficiently general to cope in a multiprocessor environment with, for example, heterogeneous processors, word sizes etc. Some of the work in hand described in the previous report has therefore been postponed pending major redesign of the Pascal Plus virtual machine.

An abstract specification of the DTL virtual machine has been designed [15]. This work has been based on the practical implementations described in the previous section supported by the more formal approaches taken in [4, 14]. It is intended to be used as the basis of a clean architecture intensive implementation of DTL which is configurable to single or multiprocessor, closely or loosely coupled architectures. It will also form a suitable vehicle for evaluation of process to processor allocation strategies being investigated by Ioannis Samaras for his PhD thesis.

Work on architecture insensitive virtual machine design is also being pursued in the Pascal Plus context by PhD student Tony Tye. This has led to the conclusion that the Belfast portable Pascal Plus virtual machine is not sufficiently general to form the basis of a fully distributed virtual machine. As a result, the work on the UCSD extended Pascal Plus has been reorganised. An architecture insensitive UCSD Pascal Plus virtual machine is being designed and implemented in Pascal. This obviously runs extremely slowly when the Pascal is itself interpreted but provides a suitable experimental vehicle for the incorporation and evaluation of the sort of parameterisation necessary for machine insensitivity. It will be used to generate assembly coded versions when the design if finalised. At the same time the problems of compatible compiler development are being tackled in a number of stages, the first of which is nearing completion and the remainder are expected to be completed during the remainder of the grant period. The first problem is the difficulty of recompiling the compiler because of its size. The short and longer term solutions to this are respectively cross compilation and compiler segmentation. The development using cross compilation, of a compiler which will compile segmented programs is already well advanced. This will then run on the virtual machine described above and recompile itself.

It is planned that further stages providing a Pascal Plus coded operating system and compiler will occupy the remainder of the grant period and will also form a sound basis for research to be undertaken on the recently awarded contract 'A Cross Development Environment for Distributed Program Development' [5].

Delivery of the UMIST provided four LSI-11/23 processors with a large disc and fast printer in support of this project was delayed until mid-June. This equipment will be complementary to CYBA-M. The combination of these two multiprocessors will provide a powerful, heterogeneous multiprocessor which will be an ideal testbed for further work on architecture insensitive virtual machines [8,13]. The development of an equally flexible development environment for distributed programs will be continued under the new grant [5].

References

1. J.W.Hughes and M.S.Powell, 'Program Specification using DTL', *Proceedings of the 1981 Aarhus Workshop on Program Specification* in Springer-Verlag lecture notes on Computer Science, 134 Program Specification.
2. D.Coleman, J.W.Hughes and M.S.Powell, 'A Method for the Syntax Directed Design of Multiprograms', *IEEE Trans on Software Engineering*, March 1981.
3. J.W.Hughes and M.S.Powell, 'A Distributed Line Monitoring System', SERC Cooperative research proposal, April 1981.
4. J.W.Hughes and M.S.Powell, 'A Reduction Specification for DTL', *Proceedings of the 1981 Aarhus Workshop on Program Specification* in the Springer-Verlag lecture notes on Computer Science, 134 Program Specification.
5. J.W.Hughes and M.S.Powell, 'A Cross Development System for Distributed Computing Systems', SERC research proposal, September 1982.
6. P.Jones, 'Joint UCSD/Pascal Plus Virtual Machine', UMIST Computation internal report, November 1982.
7. D.W.Bustard, 'The Pascal Plus Pseudo Machine', Department of Computer Science, The Queens University of Belfast, version 2.
8. J.W.Hughes and M.S.Powell, 'A General Multiprocessor Pascal Virtual Machine', UMIST Computation internal report.
9. J.W.Hughes and M.S.Powell, 'A Programmers Requirements for Parallel Languages', Loughborough Closely Coupled Systems SIG, April 1982.
10. S.Karimi, 'Implementation of Concurrent Pascal on the Cambridge Ring', UMIST Computation Department report, May 1982.
11. F.R.A.Al-Mendhry, 'A Distributed Pascal Plus Implementation', UMIST Computation Department report, April 1982.
12. A.J.R.G.Milner, 'A Calculus of Communicating Systems', Springer-Verlag Lecture Notes on Computer Science, Vol. 92, 1980.
13. K.C.B.Tan, 'An Adaptable Pascal Plus Virtual Machine Architecture for Multiprocessor Environments', UMIST Computation Department M.Sc. Thesis, 1982.
14. K.Hosokawa, 'A Formal Semantics of a Language for the Description of Communicating Systems', UMIST Computation Department PhD Thesis, 1983.
15. J.W.Hughes and M.S.Powell, 'The Implementation of DTL', accepted for publication by Software - Practice and Experience, June 1983.
16. J.W.Hughes and M.S.Powell, 'A Language for the Design and Implementation of Concurrent Programs as Structured Networks, accepted for publication by Software - Practice and Experience, June 1983.

Newcastle

DR P E LAUER
THE UNIVERSITY OF NEWCASTLE UPON TYNE

A COMPUTER BASED ENVIRONMENT FOR THE DESIGN AND ANALYSIS OF HIGHLY PARALLEL AND DISTRIBUTED OPERATING SYSTEMS
July 82 - June 84

Background, General Goals and Achievements

The advent of LSI and VLSI technology and microprogramming techniques has greatly increased the number of choices a digital system designer has for decomposing a system into ultimate subsystems which grow considerably more powerful and complex in function as technology progresses. Furthermore, designers have been learning how to combine such subsystems in new ways giving rise to systems which perform the same general functions as earlier systems but with a much greater degree of parallelism and distribution.

Concurrent systems are more difficult to specify and analyse than sequential ones, because they require the conceptualisation not only of their sequential subsystems, but also of the complex interactions between them. It follows that the programmer's intuition is not enough, being unreliable in cases of high complexity. Here solution of the problem of verification of correct behaviour of the design becomes crucial, and a satisfactory conceptual apparatus for rigorous verification is essential.

If a combination of subsystems is to cooperate coherently to perform a particular system function, synchronisation is necessary to ensure proper joint behaviours of subsystems with respect to any parts of the system they share. If the resulting system is distributed and capable of parallel behaviours in subsystems, then the required synchronisation must be specified without recourse to either a central clock, central control or global state. Many of the conventional tools the system designer has at his disposal, such as Automata Theory and Formal Language Theory, are only suitable for adequately expressing sequential systems and they do this under the assumption of a global system state. Hence, designers have had to support such formalisms with additional formal and informal notations when applying them to the specification and analysis of concurrent and distributed systems.

The project is developing and investigating a formalism (COSY from concurrent system) extending the conventional notations of automata theory and formal language theory. This extension permits one to formally express and analyse synchronisation of concurrent and distributed systems.

The conventional notions which COSY extends have been traditionally used as a basis for automating the specification and analysis of sequential and centralised systems. The extended notions of COSY are equally suited for automating the same for the concurrent and decentralised case. The project has developed a computer based environment for the COSY formalism called BCS for basic COSY system.

Specific Achievements and Results for the Year

The Basic COSY System has been refined and further developed throughout the year [6]. Incorporation of a central Command Interpreter Class has made it easier to define extensions to the command repertoire, and ensured that the command language syntax is the same for all the different dialogues that the system allows. The Environment Class provides a means of version control, by organising systematic names for the files in which descriptions of a concurrent system are kept.

The internal representation of a Basic COSY program has been revised to make possible a more direct implementation of the COSY semantics. The functions of the concurrent simulator have been modularised and extended; there are two styles of translation selection, two styles of transition echo (immediate trace of firing), and four styles of replaying the history of a system. Extending any of these choices is now a fairly straightforward programming task.

Some effort has been devoted to improving the portability and clarity of the program. An ICL Perq was delivered to the project in November 1982; unfortunately it has not proved practicable to install the Basic COSY software on the Perq, nor will it be until SIMULA is available on that machine. But the use of the Perq as a sophisticated terminal to the IBM/370 has brought more benefits than might have been expected - the production of lucid and well laid out coding has been made much more efficient by the facility to run a screen editor displaying 48 lines of program text (compared with 20 on a typical VDU), and by use of a mouse to speed the selection of text; paper consumption has been considerably reduced.

Dr Lauer has continued to make extensive use of BCS and has just completed a detailed and self-contained users' introduction to the BCS environment [2]. The document consists of three main parts; the first is the users' guide to the design environment; the second explains the syntax and formal semantics of the COSY specification notation; and the third contains a detailed analysis of a series of examples developed by means of the COSY environment. The latter serves as a vehicle for indicating to the user how he might develop this own type of analysis within the environment.

BCS has also been used both by the third year honours students in connection with their formal semantics of programming languages course, and by CSSD students and PhD students in connection with their lectures on concurrency and Petri Nets.

During the past year Dr Lauer has continued to work on some of the broader issues involved in producing self-documenting programming environments which has led him to suggest the notion of computer system dossier as a means of supporting such environments. Dr Lauer was invited to lecture at a conference on Synchronisation, Control and Communication in Distributed Computing Systems organised by the Polytechnic of Central London and INRIA. His contribution will appear as part of a book to be published by Academic Press [1].

During the period, Dr Lauer has also concerned himself with a number of more theoretical aspects of concurrency in general and the COSY formalism in particular. This activity was greatly enhanced by collaboration with visitors to the project, in particular Professor R Janicki of the Institute of Mathematics, Warsaw Technical University; and Professor R Devillers of the Universite Libre de Bruxelles, Belgium. The three individuals have separately and jointly authored a number of papers on the maximally concurrent evolution of non-sequential systems and they will be reporting on their work at the Fourth European Workshop on Applications and Theory of Petri Nets, Toulouse, France, September 1983 [4].

Finally, Dr Lauer, Professor Janicki and Dr J R Just, the latter at the Institute of Electronics Fundamentals, Warsaw Technical University, have been collaborating on the use of Petri Nets in the description of hardware configurations. They are reporting on their work in a joint paper, a shorter version of which will appear in the Proceedings of the International Conference on Microprocessors and Microprogramming, Budapest, Hungary, 1983 [3].

Newcastle

A visiting graduate student, Detlef Hillen has been concerned with the design and verification of concurrent systems. He considered rules which change the structure of such systems but preserve some dynamic properties such as adequacy and periodicity. He introduced five classes of such rules and investigated the applicability of these rules [5].

References

Items identified with an ASM number are project memoranda available from Dr Lauer

1. P.E.Lauer, 'Computer System Dossiers', in *Synchronisation, Control and Communication in Distributed Computing Systems*, Academic Press (to appear).
2. P.E.Lauer, 'User Introduction to BCS, A Computer Based Environment for Specifying, Analysing and Verifying Concurrent Systems', ASM/107, February 1983.
3. P.E.Lauer, J.R.Just and R.Janicki, 'On the Description of Simple Microprocessor Configurations by Means of Petri Nets', *International Conference on Microprocessors and Microprogramming* Budapest, Hungary, 1983.
4. P.E.Lauer, R.Janicki and R.Devillers, 'Maximal Concurrent Evolution of Non-Sequential Systems', *Proceeding of Fourth International Workshop on Applications and Theory of Petri Nets*, Toulouse, France, 1983.
5. D.Hillen, 'Adequacy-Preserving Substitution and Reduction Rules', ASM/108, June 1983.
6. P.E.Lauer, B.C.Hemshere, 'Project on a Computer Based Environment for the Design and Analysis of Highly Parallel and Distributed Computing Systems: Interim Progress Report', ASM/112, June 1983.

DR I MITRANI
THE UNIVERSITY OF NEWCASTLE UPON TYNE

MODELLING AND PERFORMANCE EVALUATION OF DISTRIBUTED COMPUTING SYSTEMS
April 81 - March 83

The complexity of distributed computing systems makes the task of assessing and predicting their performance a non-trivial one. An efficient way of approaching that task is by the construction and analysis of mathematical models. The nature of these models, as that of the systems they represent, is probabilistic.

This project set out to examine two aspects of distributed computing systems. One concerns the performance of local area networks - bus networks and ring networks in particular. The other problem has to do with the benefits that can be gained from multiprocessing: to what extent can the execution of a program be speeded up by using many processors instead of one. Approximately two thirds of the project effort (the entire first year, and a few months of the second year) were devoted to the network investigation. Some of this was carried out in collaboration with colleagues from the University of Paris-Sud, namely Professor E Gelenbe and Dr Brigitte Plateau. Two papers have been published [1,2], on Ethernet and the Cambridge Ring, respectively; a third paper - on a token ring network - will be prepared when certain simulation and numerical results are obtained.

A study of program execution by many processors has benefitted from the collaboration with Dr G Fayolle, of INRIA. A paper describing the model and some theoretical results has been prepared [3].

References

1. E.Gelenbe and I Mitrani, 'Control Policies in CSMA Local Area Networks: Ethernet Controls', in *Performance Evaluation Review,* Vol 11, No 4, pp233-240, 1982.
2. P.J.B.King and I.Mitrani, 'Modelling the Cambridge Ring', In *Performance Evaluation Review,* Vol 11, No 4, pp250-258, 1982.
3. P.J.B.King, G.Fayolle and I.Mitrani, 'On the Execution of Programs by many Processors', submitted for publication.
4. P.J.B.King, I.Mitrani and B.Plateau, 'Modelling a Token Ring Network', Research Report 12, ISEM, University of Paris-Sud,1983.

PROFESSOR B RANDELL
THE UNIVERSITY OF NEWCASTLE UPON TYNE

THE RELIABILITY AND INTEGRITY OF DISTRIBUTED COMPUTING SYSTEMS
Aug 81 - Aug 84

Background and Nature of Research

The project is a continuation of a broad programme of research into the design of computing systems that can operate continuously and satisfactorily, despite suffering from faults. Distributed computing systems have some obvious advantages over conventional systems, with respect to the possibility of providing some forms of fault tolerance. However they can also suffer from reliability problems that do not occur in conventional systems - for example, one part of the system may be taking decisions based on data that another part has already found to be erroneous.

Research Goals

The overall goal is to investigate general techniques for designing distributed computing systems which can cope effectively with situations in which it is belatedly detected that erroneous data has been spreading from computer to computer within the system. Its short term goals are mainly focussed on the incorporation of the various fault tolerance techniques the project has developed to date into a distributed system based on the UNIX operating system. Its long term goal is to gain a deeper understanding of fault tolerance and fault avoidance applied to the design of both software and hardware.

Achievements

The project has developed, and documented in a number of books and papers, general principles and practical techniques for achieving fault tolerance. Most recently it has developed a software subsystem, called the Newcastle Connection, which can be added to each of a set of physically connected UNIX or UNIX-lookalike systems so as to construct a distributed system which can be programmed like an ordinary UNIX system, thus hiding the problems normally associated with using distributed systems. The first implementation of this 'UNIX United' system was constructed at Newcastle on a set of PDP-11 computers connected by a Cambridge Ring. The Connection software is now being distributed to other organisations, and has already been used to construct UNIX United systems based on PDP11s, Motorola 68000s, VAX/750s, and ICL PERQs, variously using Cambridge Rings or Ethernets. Prototype extensions and additions to the system, for incorporating triple modular redundancy, stable storage, multi-level security and discless workstations have been completed, and various others are in progress. This work in fact constitutes a first series of explorations of a very powerful method of system implementation that has been developed, which is characterised by the use of recursive structuring.

Work in Hand

The currently available version of the Newcastle Connection is being extended so as to permit the construction of large scale UNIX United systems based on multiple local and/or wide area networks. Work is continuing on the incorporation of various fault tolerance schemes. More long term investigations are concentrated on general naming issues, and on the constructing of forward as well as backward error recovery in distributed systems.

Applicability

Many companies are actively interested in using and/or marketing systems incorporating the Newcastle Connection. The exploitation activity is being coordinated by the Newcastle-based Microelectronics Applications Research Institute (MARI), who are also responsible for distribution of the software at a nominal fee, to organisations eligible to obtain educational licences. To date MARI have negotiated commercial licences, in some cases for quite large numbers of systems, with various organisations including CERN (Switzerland), DEC (USA), ICL (UK), Research Triangle (USA), PCS (W.Germany), Portable Software Inc. (USA), SG2 (France) and Siemens (W.Germany), as well as a considerable number of educational licences. The University and MARI are also involved with several companies in major applied research and development projects that take advantage of the Newcastle Connection, and the University's work on fault tolerance and system security. These include a pilot Information Exchange System for the EEC ESPRIT Project, and a Distributed Secure System for the Ministry of Defence.

Further Information

Contact name: Brian Randell. However queries on both commercial and educational licencing arrangements for the Newcastle Connection should be directed to:

Paul Robinson
MARI
22 Jesmond Road
Newcastle upon Tyne NE2 4PQ
Tel: Newcastle (0632) 817861

References

1. T.Anderson and P.A.Lee, *'Fault Tolerance: Principles and Practice'*, Prentice-Hall 1981.
2. D.R.Brownridge, L.F.Marshall and B.Randell, 'The Newcastle Connection - or UNIXes of the World Unite', in *Software Practice and Experience*, Vol. 12, No 12, December 1982.
3. J.M.Rushby and B.Randell, 'A Distributed Secure System', *IEEE Computer*, Vol. 16, No. 7, pp. 55-67, July 1983.
4. B.Randell, 'Recursively Structured Distributed Computing Systems', in *Proceedings of the Third Symposium on Reliability in Distributed Software and Database Systems*, IEEE, Florida, October 1983.

DR P HENDERSON
UNIVERSITY OF OXFORD

FUNCTIONAL OPERATING SYSTEMS
Oct 81 -Sept 83

The project is an investigation into the application of purely functional programming languages, in particular Henderson's Lispkit Lisp [1], to the design and implementation of computer operating systems. It is intended that such operating systems will be physically distributed over a small collection of microprocessors, each microprocessor providing an independent and well defined service within the operating system. The collection of processors will be integrated by a simple communications network to give a powerful single system. Functional programming languages are good candidates for designing such systems for several reasons. They are well suited to the task of describing the behaviour of independent processes which communicate with their neighbours by message passing along serial channels. Their mathematical simplicity and rigid programming discipline enables rapid developments of reliable software.

The investigations in this project have yielded an understanding of the functional programming style in an important field of applications. Operating systems are clearly a vital component of future computer systems, and the application of functional programming should provide indications as to how such systems may be simplified in structure, and hence be developed more easily, and made more reliable. In order to focus attention it is intended that a long term goal will be to implement, in a functional style, a complete system to support a programmer who wishes to develop functional programs. Research on this topic is continuing with support from ICL.

Several prototype mini-operating systems have been developed in Lispkit Lisp which execute interactively on the single processor implementation described above. These range from simple systems described by Henderson [2], and other similar systems designed around the concept of communicating processes, to rather different systems inspired by the Unix concepts of 'shells' and 'pipes'. These experiments suggest that there are many more interesting operating systems to be designed which are at the same time simple yet very powerful. The experiments are also forcing a detailed consideration of techniques for handling important problems such as nondeterminism, concurrency, and processes with multiple input and output streams. This experience has been an essential contribution to the redesign of the Lispkit Lisp implementation which now enable it to support the operating system components executing at each node of the microprocessor network.

In order to be able to define operating systems which execute a number of concurrent processes it was necessary to extend the programming language Lispkit Lisp with a non-deterministic form of expression. The basic form of this expression as currently used allows two concurrent processes to be commenced and for the termination of one of them to discard the other. This is adequate to define the sorts of concurrency needed to be able to define even quite general multi-programming environments. The process of deciding how this language primitive can be packaged for presentation to more general users is still being resolved. It is fully described in [3]. The necessary extensions to the abstract machine are reasonably straightforward. This is a result of some significance, and one which the group intend to take fullest advantage of in subsequent research.

A second operational problem which has been solved concerns the need for communication among functional programs when a network of processors is involved. Simple interactive functional programs are defined as functions from one input stream (the keyboard, say) to one output stream (the screen, for example). When a functional program must communicate with more than one user, or with a mixture of users and other machines, it is necessary to have many input and output streams. Various solutions were investigated. Finally it was decided upon, and has been implemented, a scheme whereby the program is a function from a tuple of streams to a tuple of streams. The details of this solution are also reported at length in [3] as are the necessary extensions to the abstract machine.

Armed with these completed developments the project is now into the final stage as originally proposed, being able to demonstrate a simple two machine file transfer system using two Perqs back to back using an RS232 link. These experiments are in the process of being repeated using other machines available in greater multiplicity. The results of these experiments along with others carried out on a single processor will be reported on shortly. Experiments on the single processor have been to demonstrate that Lispkit Lisp is adequate for writing operating system shells, notwithstanding that it is a purely functional language.

In this project concentration has been on the provision of basic machine and language facilities for defining operating systems. The group have become increasingly aware that, while they have validated their conjecture that they can write operating systems in purely functional style there are many other applications which would benefit from similar developments. It is believed that a system has been produced which will allow the investigations to move on to study the use of functional programming for a wider range of applications.

In order to encourage and promote the study of functional programming the Lispkit Lisp implementation and associated software tools are available for distribution to interested researchers. The implementation is available for PERQs, for Intel 8086 (or Intel 8088) based computers, for Motorola 68000 based computers, and for the VAX [4]. In addition a reference implementations in Pascal and C are available.

References

1. P.Henderson, *'Functional Programming: Application and Implementation'*, Prentice-Hall International, 1980.
2. P.Henderson, 'Purely Functional Operating Systems', in *Functional Programming and its Applications,* Cambridge University Press 1982.
3. S.B.Jones, 'Abstract Machine Support for Purely Functional Operating Systems', Programming Research Group Technical Monograph, PRG-34, Oxford, 1983.
4. P.Henderson, G.Jones and S.B.Jones, 'The Lispkit Manual', (two volumes) Programming Research Group Technical Monograph, PRG-32, Oxford, 1983.

PROFESSOR C A R HOARE, J E STOY and M K HARPER
UNIVERSITY OF OXFORD

DISTRIBUTED COMPUTING SOFTWARE
Jan 82 - Dec 84

The availability of inexpensive computing power has made it economically attractive to construct computer systems from physically discrete interacting components. Such distributed systems require

1. a communication medium;
2. a methodology for the construction of distributed software.

The Oxford Distributed Computing Project addresses the second requirement. The latest software engineering techniques are used to aid in specification and design, and to avoid preconceptions inherent in traditional computing environments which are less relevant for modern technology.

Formal specification provides a framework in which software designs may be proposed and partly evaluated in advance of (even a prototype) implementation. The project uses a specification technique drawn almost entirely from classical mathematics; this allows great freedom in the exploration of new software designs. It is intended that the distributed system be accompanied by a specification which can be published independently of its implementation, and that the specification be precise and clear enough to allow, for example, independent supply of the system's components (interworking). The specification could also serve as an effective national or international standard of high quality.

Progress has been made in several areas. Firstly, an experimental device service [1] has been provided which acts as a distributed extension of the UCSD P-system. Secondly, experience has been gained by the project staff with specification techniques [2] and some progress has been made in improvement of them. Thirdly, the design of classes of distributed system service has been explored and implementable services have been specified. Two services in particular, one for providing low-level disc storage [5], and the other for providing spooling to a printer [6], have been both specified and implemented. Further work has been done on the rigorous specification of communication protocols.

In the coming year, the provision of services will be extended towards a higher-level filing system and the mail facilities (see [3,4] for example specifications of services of these kinds), as well as archiving, logging and accounting services. There is much collaboration in matters of specification with members of the Software Engineering project, and good contact with visitors from industry.

The eventual publication of a distributed system specification will make the design's applicability largely hardware independent; conformance to the design will allow interchangeability of components (whether supplied by manufacturer or by user).

For further information contact Dr C Morgan or Dr R Gimson.

References

1. A.Newman, 'An Experimental UCSD Device Service for the Cambridge Ring', Programming Research Group, February 1982.
2. C.Morgan, 'Specification of a Communication System', to appear in *Proceedings of the International Conference on Synchronisation, Communication and Control in Distributed Computer Systems,* Polytechnic of Central London, 20-24 September 1982, Academic Press.

3. B.Sufrin, 'Formal Specification of an Automated Mail System', Programming Research Group Internal Report, April 1982.

4. C.Morgan and B.Sufrin, 'Specification of the UNIX Filing System', to appear in *IEEE Trans. Software Engineering*.

5. 'A Block Storage Service', Programming Research Group Internal Report, May 1983.

6. 'A Printing Service', Programming Research Group Internal Report, July 1983.

DR J D ROBERTS
UNIVERSITY OF READING

PROGRAM TRANSFORMATIONS FOR TIGHTLY COUPLES DISTRIBUTED SYSTEMS
March 82 - Feb 85

Although detailed studies during 1982-83 have been confined to fairly simple examples, it is possible to see two distinct roles emerging for transformations of distributed programs.

The first is in the theoretical derivation of fundamental programs (library programs). This is exemplified by the systematic translation of a Lucid specification to a Dataflow diagram and thence to a collection of C.S.P's (communicating special processes). It is viewed primarily as a human reasoning activity.

The second is in the transformation of programs expressed in a neutral, configuration-independent and typically very highly distributed form into some other representation more suitable for given actual configurations. The main transformations here are the lumping together of specified groups of processes into single processes and also the multiplying of parallel channels into a single channel. This is viewed as an activity to be performed automatically under manual direction, and explicit algorithms have been identified for restricted cases.

One particular case which is highly restrictive, but nevertheless sufficiently applicable to be of a practical use, is the pipeline (ie. any system in which no data is recirculated and reviewed as a collection of C.S.P.'s which after an initialisation period enter a simple loop) [1]. In this case groups of C.S.P's can be merged to produce similar C.S.P's without problems of deadlock nor any other complication; but if greater generality is allowed, the class does not appear to be 'closed' under this type of transformation until schemata are considered of very great generality. The current theoretical work is for this reason directed towards identifying automatable transformations for these general situations, Petri nets being used as representations. It is expected that programs resulting from this process would be more suitable for automatic compilation than human comprehension.

As always, the investigators would be pleased to communicate with others regarding any potential applications especially in real-time data or signal processing.

References

1. J.D.Roberts and S.A.Williams, 'The Construction and Transformation of Collections of Communicating Sequential Processes that Represent Pipeline Configurations', Internal Report RCS 181, Nov. 1983.

DR D H BELL, N WILLIS and DR J M KERRIDGE
SHEFFIELD CITY POLYTECHNIC

AN EVALUATION OF ALTERNATIVE IMPLEMENTATIONS OF MULTIPROCESSOR COMMUNICATION MECHANISMS
March 82 - Feb 85

Description

The aims of the research are :

1. to understand the nature and types of interaction that can take place in computer systems that involve process to process communications and in particular those involving loosely coupled micro-computers.
2. to determine the relative merits of Pascal Plus and Edison in terms of their ease of use in describing systems involving communication processes. Pascal Plus and Edison have been chosen because these are recent languages that incorporate the communication mechanisms of monitors and distributed processes respectively. They are also languages for which transportable implementations are available.
3. to assess the relative merits of Pascal Plus and Edison on the criteria of their memory requirements, speed and cost of implementation of the run time system for the communication mechanisms.
4. to determine the best way of implementing the Pascal Plus and Edison communication mechanisms when computers are linked using the following hardware :

   Cambridge Ring
   Econet
   Point to point using parallel lines

The results of the investigation will be :

1. evidence of how local industries could use Pascal Plus and Edison;
2. qualitative information on the relative ease of use by programmers of Pascal Plus and Edison;
3. data on the performance overheads associated with using these languages for implementing multi-processor based systems;
4. data describing the costs of construction of the Pascal Plus and Edison communications techniques.

The emphasis of this work lies in the implementation of the run-time system for the languages. This is because we believe that there are immediate problems in this area which must be solved before the language ideas can be successfully exploited.

Pascal Plus and Edison have been chosen because they are typical of concurrent approaches to the problems outlined.

Programme

Survey of Local Industry

This part of the project aims to keep our feet firmly on the ground by:

1. finding out what practical use is made of parallelism and distributed computing;

2. locating applications that we could use as test vehicles for our systems;
3. seeking commercial applications for such work.

We have had talks with three organisations - a health authority implementing an information system, a large nationalised manufacturing company, and a major manufacturer of office automation equipment. Our main findings have been that parallelism is, in practice, not something that the applications programmer is generally concerned with; and no concern about problems of implementing software system involving parallelism. As a result we feel that probably the area of real time embedded computer systems will be more fruitful.

Ease of Use Experiment

Which notation for describing interactions between parallel processes is the easiest to use? This is the question that this project seeks to answer using the methods of experimental psychology.

Typically an experiment involves a group of people who are split into two. One group is asked questions about program text. The time taken is recorded. The other group similarly examine a different program text and answer questions about it. The two programs solve the same problem but with different notations. The aim of the experiment is to measure ease of *comprehension*. Comprehension is chosen because it is considered vital for program design, debugging and maintenance. We plan to compare the two fundamental mechanisms - procedures and monitors.

Experimental design is at an advanced stage. Issues include the size of the subject population, their background, character of the problems, nature of the questions. A working paper is available. We plan to carry out a pilot experiment before the end of 1983 and a full experiment by June 1984. We are fortunate in having the collaboration of workers in the MRC/SSRC Social Psychology Group at Sheffield University (Thomas Green et al) who have considerable experience in experiments of this sort.

Pascal Plus

Version 1c of the Pascal Plus system (made available by Queens University, Belfast) has been implemented. A Fortran version of the Pascal Plus compiler (P-code generator) has been written on the Polytechnics IBM 4341 enabling Pascal Plus programs to be developed on the mainframe. After compilation the resultant P-code can be downloaded to a Z80 based Ithaca microcomputer on which resides the interpreter part of the Pascal Plus system.

What remains to be done once we have uniprocessor implementation of the language working, is to reimplement relevant parts of the compiler and interpreter to enable actual multiprocessor execution (on several Ithacas connected by a Cambridge Ring, Econet and point to point lines) of Pascal Plus programs.

Edison

Conversion of the Edison kernel into Z80 assembler code is now at an advanced stage. The remaining conversion work to be done is in providing the input/output routines for the Z80 processor. The operating system is written in Edison and the Kernel has to be capable of interpreting this before running other programs. Thus, the first real 'live' test will be execution of the operating system.

DR D G SMITH
UNIVERSITY OF STRATHCLYDE

OVERLOAD CONTROL IN PROCESSOR-BASED SYSTEMS
Feb 83 - Jan 84

Aims

This project aims to investigate methods of controlling the behaviour of processor-based systems under overload. An algorithm will be studied which observes the system and controls the degradation of service as the loading increases.

Background

In any information processing system there is always the problem of assigning resources of the system to users who are demanding service in a satisfactory fashion. The questions of what resource assignment policies one can adopt, and of what is the measure of satisfaction by which they should be compared depends, on the design of the information processing system and the purpose that it was designed to serve. But it seems likely that similar techniques might be used to study quite a variety of structures, and that it would be possible to construct a kind of theory of overload control by considering the common aspects of a number of different problems.

There are two broad areas into which the field can be divided, these being data transmission networks and data processing systems. In the transmission networks principal concern is with the transmission and reception of information, and, while it may be necessary to examine some of the information and do processing work on it at the nodes, this is incidental to the main function of the network which is the transmission. In the data processing system the main purpose is information processing, which is undertaken to achieve some end in computing or real time control. There may be transmission of information over various physical facilities in the course of performing the processing function, but this time the transmission is incidental to the main function of the system.

Both kinds of structure need some sort of load control. At the present time, more attention has been given to flow control in a data transmission network than to load control in an information processing system. But neither application has yet produced a well founded framework of definitions and objectives in terms of which the load control problem can be discussed.

Present State of the Art

The information processing function which the investigator has in mind is the control of an exchange in an SPC telephone system. There are a number of such systems currently operating throughout the world. Each different system has a different structure for the processing, and probably they all have different load control arrangements heuristically designed in a way which will, (it is hoped) to some extent meet a variety of objectives which are judged to be desirable.

The situation with data-networks is rather different. The objectives are clearer, namely to avoid any kind of lockout or crash, while minimising the response time of the network to accepted traffic. Load control methods have been in use and under discussion for some time, with a view to achieving these ends.

PROFESSOR R L GRIMSDALE and DR F HALSALL
UNIVERSITY OF SUSSEX

THE DESIGN AND IMPLEMENTATION OF A MULTIPROCESSOR SYSTEM
Oct 79 - March 84

Overview and Aims

This project is concerned with the design and implementation of a multiple microprocessor computer system. The main aims of the project are firstly to obtain a better understanding of the problems arising across the hardware software engineering boundary within multiple processor structures and secondly, to develop a methodology for implementing applications software for such systems.

Work to Date

The hardware structure adopted is comprised of a number of multiprocessor-shared memory 'stations' which are linked together by means of a high-bandwidth serial communications sub-network. Both the number of processors within a station and the number of stations within the system can be readily varied and hence the implemented system is useful for investigating a wide range of different application environments.

The software being developed for the system is made up of a number of components. These include software tools to aid program development and testing (for example compilers, interpreters and performance monitoring and diagnostic software), system generation software to aid the mapping of applications software onto the selected multiple processor structure, and run-time support software for implementing intertask communication on the target hardware.

The laboratory development facility is comprised of the selected target multiprocessor hardware structure front-ended by a separate host machine. The latter contains both the language compiler and system configuration software and also the performance monitoring and diagnostic software. The host is interfaced to the same communications sub-network as is used for interstation communication within the multiprocessor. This provides the users of the system with a convenient mechanism for investigating and monitoring alternative hardware and software assignment strategies. In order to make the laboratory development facility usable for a range of applications, most of the system is constructed from commercially available hardware components. Thus each station is comprised of a number of Intel 8086 based single board computers and the interstation communication sub-network is based on Ethernet.

The application programming language which has been developed is known as Martlet. It is based on two languages: Pascal and Ada. The sequential part of the language is basic Pascal and this has been extended to include the Ada commands to control and synchronise intertask communication. Wherever possible, all other supporting software is written in Pascal. The laboratory development facility is now operational and is currently being used as a research vehicle for investigating a range of problem areas associated with the use of multiprocessors in a number of real-time embedded applications.

Further Information

A number of papers and conference contributions have been written describing aspects of the project and a selected list follows. For further information please contact either Professor Grimsdale or Dr Halsall.

References

1. R.L.Grimsdale, F.Halsall, F.Martin-Polo and G.C.Shoja, 'POLYPROC: The University of Sussex Multiple Microprocessor System', *IEEE 2nd International Conference on Distributed Computer Systems*, Paris, April 1981.
2. R.L.Grimsdale, F.Halsall, F.Martin-Polo and S.A.Wong, 'Structure and Tasking Features of the Programming Language Martlet', *Proceedings IEE*, Vol. 129, Pt. E, No. 2, March 1982.
3. G.C.Shoja, F.Halsall and R.L.Grimsdale, 'Control Kernel to Support Ada Intertask Communication on a Multiprocessor Computer', *IEE Journal Software and Microsystems*, Vol. 1, No. 5, August 1982.
4. G.C.Shoja, F.Halsall and R.L.Grimsdale, 'Some Experiences of Implementing the Ada Concurrency Facilities on a Distributed Microprocessor Computing System', *Software and Microsystems*, Vol. 1, No. 6, October 1982.
5. F.Halsall, R.L.Grimsdale, J.E.Lambert and G.C.Shoja, 'Development Environment for the Design and Test of Applications Software for a Distributed Multiprocessor Computer System', *Proceedings IEE*, Vol. 130, Pt. E, No. 1, January 1983.

PROFESSOR R L GRIMSDALE and DR F HALSALL
UNIVERSITY OF SUSSEX

DECENTRALISED RESOURCE MANAGEMENT IN A DISTRIBUTED MULTIPROCESSOR COMPUTER SYSTEM
Jan 82 - Dec 84

Overview and Aims

The aim of this project is to investigate the design and implementation of the operating system software required for the management and control of the shared resources - both hardware (files, peripheral devices, etc.) and the software (shared data structures, schedulers, etc.) - in a distributed computer system which provides resiliency to both hardware and software failures. It is envisaged that the proposed software structure will be particularly suitable for those application areas of multiprocessors such as on-line terminal enquiry systems which demand a high level of system availability.

Work to Date

To satisfy the requirements of these systems, resiliency in the system software is provided by the avoidance of any single, critical, centralised controllers. Instead, control is distributed throughout the processing units which comprise the system. Thus loss of a processing unit through failure does not result in a total system failure but rather the loss of some localised processing function and resource.

The system software is organised as a suite of virtual resource rings, each ring managing all resource components of a particular resource type in the system. A ring consists of a set of instances of a software task known as a ring element. These instances in turn control access to specific components of the resource type managed by that ring. Each element in a ring then receives calls - requests for a resource or output from a resource - from its preceding element in the ring and in turn makes calls to its succeeding element. Thus in order to construct a ring, it is only necessary for each element to know the identity of its succeeding element in the ring.

The loss or failure of a processing unit within the system will be reflected in each ring for which is contained an element. This loss is detected by each failed element's predecessor in the ring as a failure to obtain a good reply to a call and hence the predecessor automatically tries to establish a new communication path to the next available element. Thus a faulty system processing unit is automatically isolated from all the affected rings (and hence the system) and the rings themselves are automatically reconfigured to reflect the loss.

The ability to reconfigure the system, to add additional resources as demand rises or to remove processing units for preventive maintenance, means the system has good modularity and extensibility properties.

The research is being conducted using the facilities currently available on the University of Sussex multiple microprocessor system Polyproc [1]. The implementation of Polyproc forms part of the current DCS programme and although the facilities provided by it are currently being enhanced, the basic system is now operational and this is proving to be an ideal vehicle for an investigation of this type. All application software for Polyproc is written in a high-level structured language - Martlet - and the system provides a convenient facility for both the assigning and distributing application software onto the multiprocessor and also for monitoring subsequent performance.

Further Information

A more detailed description of the virtual resource ring and its operation can be found in reference 2 below and for further information please contact either Professor Grimsdale or Dr Halsall at the University of Sussex.

References

1. R.L.Grimsdale, F.Halsall, F.Martin-Polo and G.C.Shoja, 'Polyproc: The University of Sussex Multiple Microprocessor System', *Proceedings 2nd International Conference on Distributed Computing Systems*, Paris, April 1981.
2. R.Hull, F.Halsall and R.L.Grimsdale, 'Decentralised Resource Management in a Resilient Distributed Computer System', Submitted to *Proceedings IEE Part (E), Computers and Digital Techniques*.

PROFESSOR R L GRIMSDALE and DR F HALSALL
UNIVERSITY OF SUSSEX

THE DESIGN AND IMPLEMENTATION OF A BROADBAND LOCAL AREA NETWORK
Oct 82 - Sept 85

Overview and Aims

This project is concerned with research into an alternative transmission system for use in local area networks which provides a communications facility having only modest overheads per user device but offering considerable versatility both in its use and for future development. The system is based on a frequency division multiplexed (FDM) broadband transmission facility similar to that currently employed in coaxial cable television distributed networks. Much of the technology used in such networks can, with the aid of some additional RF components (modems, head-end remodulators etc.), also be used to provide versatile transmission facility for carrying a wide range of different data traffic on a single coaxial cable. These are obtained by the allocation of different frequency channels within the total available bandwidth of the cable and associated repeater amplifiers. Whereas the majority of existing networks are strongly associated with a specific protocol to which all devices must conform, within a local area network there is also a need for peer-to peer communication with minimum constraints on the communicating devices. Thus the provision of a number of 9.6K bit/sec channels available on a long term base or on demand will be of value in meeting this requirement. Further channels at 64K bit/sec can be provided which might be used either for voice communication or as dedicated links to provide the transmission layer of a packet-switched network (X25, for example). One or more high bit rate channels of the order of 10M bit/sec capability can be dedicated to a CSMA/CD facility. The same cable will also be capable of supporting a number of video links using existing CATV technology.

This is a collaborative project with Rediffusion Engineering Limited. This company has considerable expertise in the use and installation of CATV systems and the aim, therefore, is to build on this expertise and to investigate the design of the additional system components necessary to use this technology in the field of local area networks.

Work to Date

It is proposed to construct a complete system at the University and install an number of intercommunicating devices. These will include intelligent devices (computers etc.) also a number of unintelligent terminals to test the dedicated and on-demand low bit rate channels. The experimental system will also include a head-end remodulator and performance monitoring and diagnostic facilities.

Work has commenced on the design and implementation of a low bit rate frequency-agile modem which is also suitable for use as a fixed-frequency modem. This design exploits the use of modern frequency synthesiser technology for channel selection and also a dedicated microprocessor for its control. In addition, work is in progress on the implementation of a high bit rate (10 Mbps) fixed frequency modem which is intended for use with a CSMA/CD based channel.

Initially it is proposed to implement a simple head-end remodulator to allow the two designs of modems to be tested on an actual cable and when these are firm to concentrate on the design of the remodulator itself. It is envisaged that work on the system performance monitoring and diagnostic facility will also commence at this time.

Further Information

A number of internal reports have now been produced which describe selected aspects of the above work and further information can be obtained from either Professor Grimsdale or Dr Halsall.

PROFESSOR R L GRIMSDALE and DR D J WOOLLONS
UNIVERSITY OF SUSSEX

ADVANCED TECHNIQUES FOR THE COMPUTER GENERATION OF IMAGES
May 83 - Dec 83

Overview and Aims

Research work is underway at the University of Sussex on a low cost visual system for flight simulators using computer generated images. The work divides into three main areas: the creation of and access to large data bases, the development of a surface based visual system and the development of a texture generation system.

Progress to Date

A simple prototype of the surface based system has been demonstrated and the implementation of an enlarged system is currently in hand. Ultimately this will be capable of up to 500 surfaces per frame. A test pattern has been achieved using texture generation hardware that can be flown in real time.

The integration of texture and surface based display systems, scheduled for the end of 1983, can be achieved in one of two ways. Either the texture system is superimposed on the surfaces (e.g. brickworks on walls or ploughlines across fields) or the surfaces are superimposed on a textured ground plain; the ground plain would be a pseudo random patchwork of hangers, houses, roads and mountains. Suitable database selection will enable levels of detail to be selected according to the distance of the observer.

The high speed processors needed for the generation of the images have been developed at Sussex with a design philosophy of modularity and expansibility with emphasis on high quality engineering and manufacture. Various types of modular processor have been produced using the custom 'solder wrap' technique.

Investigation is currently underway on the possibility of implementing certain parts of the circuitry in VLSI. It is hoped that a system will emerge that will be exploited commercially while further research work, such as the generation of complex objects like trees, continues at Sussex.

Further Information

A number of internal reports have now been produces which describe selected aspects of the above work, and further information can be obtained from either Professor Grimsdale or Dr Woollons.

DR J R W HUNTER, DR K BAKER and DR A SLOMAN
THE UNIVERSITY OF SUSSEX

INTERACTIVE SOFTWARE TOOLS FOR DISTRIBUTED COMPUTING SYSTEMS WITH AN APPLICATION TO PICTURE INTERPRETATION.
April 79 - Sept 82

Background and Nature of Research

The underlying assumption behind this project is that as computer hardware becomes less expensive, it will be possible and desirable to build systems consisting of many computers all cooperating in the solution of a single complex problem of the type encountered in Artificial Intelligence (AI) projects. Programming such systems will pose new and difficult problems, and the need has arisen to develop as sophisticated tools as possible to help in this task; it was felt that such tools should be interactive in nature.

Research Goals

The original goals of the project were:

1. To develop interactive tools for implementing and validating software for distributed computing systems.
2. To re-implement an existing AI picture-interpretation system (POPEYE) on the distributed system, applying the results of (a).

The hardware consisted of a PDP-11/34 'host' minicomputer and a number of LSI-11/2 distributed machines; communication was by individual serial lines. The host machine ran the UNIX operating system, and the distributed machines were to provide a UNIX-like environment for processes running on them. It was to be possible to run more than one process on the host, and to run a given process either on the host or on an LSI-11 with minimal charges.

It became evident that the serial lines were inadequate for the task, and a new goal was set of implementing the system using the recently developed Cambridge Ring for communication. In addition, it was decided that a simple re-implementation of POPEYE would not explore the possibilities opened up by the distributed system, and that a new approach to that part of POPEYE which deals with letter recognition should be developed. In parallel with the work on image interpretation, it was decided to implement distributed programs in several other AI domains (viz programming aids, natural language understanding).

Achievements

The project has produced a set of software tools which allow the user to specify a distributed system in terms of both hardware and software, and to initialise such a system correctly. The interactive AI language POP-11 is available on all of the machines, this is supported on the LSI-11's by a single process operating system called MONIX. However the use of serial lines produces a system which is too slow to be of practical use (see below).

A number of distributed programs have been implemented to test the system; these have all been run in 'simulation' on the host.

One program in fact provides an environment for the development of distributed programs; it consists of a network of processes which enable the user to inspect the design from different points of view.

A program which approaches the letter-recognition problem by analysing the input data in a series of fixations has been implemented.

Finally, a distributed parsing program demonstrates the use of interrupts in co-operative problem-solving and learning: a 'master' program sets a task which may be attempted in parallel by different 'servants', using different strategies: e.g. one general but slow, the other faster but requiring prior 'learning experience'. The first to finish may interrupt others, and send information for future use.

Work in Hand

The project finished in September 1982 without the Cambridge Ring software being operational; however one of the grant holders (Dr Hunter) in collaboration with Dr C Mellish is to be supported for a further year to complete the installation of POP-11 and to mount the logic programming language PROLOG on the distributed system.

Applicability

The system will provide a good interactive environment for investigating the decomposition of Intelligent Knowledge Based Systems.

Further Information

Contact Dr J Hunter.

References

1. A.Ramsay, 'An Environment for Developing Distributed Programs', *Proceedings of ECAI - 82,* Paris 1982.
2. D.Owen, 'Relating Object Models to Segmentation Strategies', *Proceedings of ECAI - 82,* Paris 1982.

J A SHARP
UNIVERSITY COLLEGE OF SWANSEA

A DATAFLOW PROGRAM DESIGN ENVIRONMENT - ITS IMPLEMENTATION AND USE AS A DESIGN VEHICLE
April 81 - March 83

Research Goals

This project aimed to develop a program design environment using a dataflow notation as the basis. Regarding a program as a description of a Problem Solution Environment (PSE), the program design environment is seen as a means of specifying PSEs. The dataflow model supports this in providing descriptive rather than imperative notations which are well suited to serve as PSEs.

Progress Made

An initial naive sequential demand-driven implementation using a tree structure to represent the PSEs demonstrated the feasibility of adapting a dataflow notation for specifying PSEs. Also included was a simple method of representing input and output by infinite functions.

A second implementation takes a set of function definitions and translates them into an intermediate code representation of a Problem Solution Environment. An expression for evaluation (a problem) is then translated into the same intermediate code, and fed into the workspace of one of a number of (simulated) processors (currently up to 10). Parallel evaluation can then be modelled.

The evaluation strategy is again essentially demand driven. In order to evaluate the original expression various other functions from the PSEs are 'demanded'. These functions in turn 'demand' others. A demand for a function to be evaluated results in the definition of that function being placed into a queue of 'demanded' definitions waiting to be evaluated when a process becomes free.

The system is not purely demand-driven since all the expressions which are passed as parameters to a function are 'demanded' when the function is called and are not passed through for the function to 'demand' internally only when required. This method was adopted to increase the potential for parallelism, and to a limited extent avoid the possibility of duplicated evaluation of arguments.

A simple version of this implementation is now working, and some measures on the amount of parallelism achievable in this simple model have been obtained.

W W WADGE
UNIVERSITY OF WARWICK

LUCID AS A DATAFLOW LANGUAGE
Oct 80 - Sept 83

Background

The overall aim of the project is to investigate the potential of the programming language LUCID as a serious and fairly general purpose dataflow language.

LUCID was designed by Bill Wadge and Ed Ashcroft (of the University of Waterloo in Canada). It is a purely declarative language with the property that programs can (in spite of the non procedural nature of the language) be understood as iteration or repetition (recursion is also allowed). LUCID is based on an algebra of histories, a history being an infinite sequence of data items. Operationally, a history can be thought of as a record of the changing values of a variable, history operations such as first and next can be understood in ways suggested by their names. LUCID was originally thought of as a kind of very disciplined mathematically pure single assignment language in which verification would be very much simplified. However, the dataflow interpretation has been very important in helping the direction in which LUCID has evolved.

Achievements

A LUCID interpreter (written in 'C') has been written at the University of Waterloo. A great deal of work has been contributed by this project.

A book 'LUCID the Dataflow Programming Language' is in preparation. For further information contact W Wadge.

DR C WHITBY-STREVENS and D MAY
WARWICK UNIVERSITY

A BUILDING BLOCK SYSTEM FOR DISTRIBUTED COMPUTING
Oct 77 - April 80

Background

The increased use of distributed computing systems has made the development of the new techniques for designing and programming them essential. The aim of this was to provide a set of building blocks to support a top-down design process and to experiment with various topologies for a system containing a number of processors.

Achievements

The project centred the design and implementation of a programming language called EPL, intended to be used for the implementation of distributed database systems, real-time control systems etc.

EPL is based on the idea of actors, which communicate only by sending and receiving messages. EPL separates the description of a computation from the actor which performs it. (Such a description is often called an act.) Thus many actors may be created to perform the same act (on different data). Although an act may not refer to any global variables, it may have parameters, which are initialised when an actor is created from the act, and are then treated as local variables of the actor. This provides a means to inform actors about each other's existence, thereby allowing them to communicate.

EPL allows the declaration of structures of actors, such as arrays. Once such a structure is created, all the component actors start executing simultaneously, probably exchanging messages with each other. Note that each of the actors may itself be expressed in terms of other actors. This provides a means of 'top-down' programming, since any actor can be decomposed into a set of intercommunicating sub-actors. In fact, an actor may have no existence at run-time, serving only to encapsulate a group of smaller actors.

The message sending primitives contain no buffering. Thus when an actor tries to receive a message, it waits until another actor sends it one, and when an actor tries to send a message, it waits until the target actor receives it. This provides a conceptually simple mechanism for the programmer, and it is relatively easy to implement on different architectures. However, there are arguments in favour of other schemes. For example, single buffering enables a communication system to proceed with the transmission of a message while the sender continues executing.

This project terminated in April 1980, following the resignations of Dr Whitby-Strevens and Mr May from the University to take up appointments with Inmos Ltd.

References

1. M.D.May, R.J.B.Taylor and C.Whitby-Strevens, 'EPL - An Experimental Language for Distributed Computing', *Proceedings Trends and Applications: 1978 Distributed Processing, IEEE Computer Society,* May 1978.
2. M.D.May and R.J.B.Taylor, 'The EPL Programming Manual', Distributed Computing Project No. 1, University of Warwick, May 1979.

3. C.Whitby-Strevens (ed), 'Record of a Workshop on Programming Languages for Distributed Computing; September 1978', Distributed Computing Project Report No. 2, University of Warwick, June 1979.

4. C.Whitby-Strevens, 'On the Performance of Dijkstra's Self-Stabilising Algorithms in Spite of Distributed Control', *1st Annual Conference on Distributed Computing Systems*, IEEE October 1979, pp 582-592.

5. C.Whitby-Strevens, 'Towards the Performance Evaluation of Distributed Computing Systems', *COMPSAC '78* (in expanded form in Computer Centre Report No 18, University of Warwick, August 1978).

PROFESSOR I C PYLE
UNIVERSITY OF YORK

REAL TIME PROGRAMMING LANGUAGES FOR INDUSTRIAL AND SCIENTIFIC PROCESS CONTROL

April 75 - Dec 78

The original objectives were to study requirements for real-time programming languages with particular reference to the work of the LTPL-E (Long Term Procedural Language - Europe) committee. It was intended to make trial compilers and to study various features by writing programs to assess them, paying particular attention to tasking and run-time aspects of the language.

During the period of the project, the LTPL-E committee changed its role, and the impetus of the language design for real-time systems left Europe with the establishment of the US Department of Defence High Order Language project which has now defined Ada. Also the programming language Modula was published, as a language for dedicated real-time systems.

Since the languages for the DoD project were based on Pascal, and Modula was designed in the light of Wirth's experience with Pascal, it was decided to implement a Modula compiler as a basis for the studies. Most of the studies carried out were on Modula, but also included paper programming in Ada, and possible Modula extensions. The York Modula compiler has been successful. Distribution and promulgation of the compiler was supported by a separate grant to Dr I C Wand.

Programming in Modula for a variety of projects in the Department gave a basis of experience by about ten people, from undergraduates upwards. The influence of the target machine architecture was studied, particularly for the Intel 8080, Zilog Z80 and Texas 9900 microprocessors. Work on the Z80 was done in association with Linotype Paul Limited, who are now using Modula as a system implementation language; they subsequently developed Modula compilers for the Z80 and Texas 9900.

For further information about the project contact Professor I C Pyle. For information about York's Modula compiler contact Dr I C Wand.

References

1. J.Holden and I.C.Wand, 'Experience with the Programming Language Modula', *Proceedings of the 1977 IFAC/IFIP Real-Time Programming Workshop*, editor C.H.Smedema, Pergamon Press (1978), pp 3-11 and 28-33.
2. I.C.Pyle and I.C.Wand, 'Real-Time Programming Languages for Industrial and Scientific Process Control', Computer Science Report No 21, University of York, May 1979.
3. J.Holden and I.C.Wand, 'An Assessment of Modula', *Software Practice and Experience*, Vol 10, pp 593-621, 1980.
4. I.D.Cottam, 'Functional Specification of the Modula Compiler - Release 2', Computer Science Report No 20, University of York, 1979.
5. I.D.Cottam, 'Functional Specification of the Modula Compiler - Release 3', Computer Science Report No 33, University of York.

DR I C WAND
UNIVERSITY OF YORK

AN OPERATING SYSTEM FOR A NETWORK OF PERSONAL COMPUTERS
Nov 81 - Nov 84

Background and Nature of Research

There is a trend to decentralise computing resources. As a result, several research programmes have been set up to investigate the possibility of adapting existing systems to a distributed role.

The origins of the present project lie in an earlier investigation [1] into the feasibility of distributing the popular and successful UNIX operating system among several processors. Several topologies were considered: a network of personal computers each with its own terminal and disk was finally chosen as the most suitable; the communication subsystem is used to share costly resources. This design provides the basis for the present project.

In the earlier project the original goal of distributing UNIX proper was altered to that of building an operating system which retains some of the features of UNIX. There is to be a hierarchical file system, whose structure and operation is similar to UNIX. However, the process model adopted is entirely different, as it is based on the Ada task.

The machines have a client-server relationship to each other. Each machine on the network will run a common kernel: roles are determined merely by the programs run on top of this kernel. For example, a machine may act as a printer server because it happens to have a device attached; it acts as a client to a file server in that it may request files to print.

Research goals

There are two major requirements for the projected system. One is that each machine should be capable of running 'stand-alone' without logical or physical connection to the network. The other is that the filing system should appear the same, wherever a user happens to be; although 'personal computers' have been specified, users will be able to use machines other than their own, albeit with some loss of privilege. The personal computers are to be connected together by a high-speed local area network, in this case the Cambridge Ring [2].

The prime reasons for the use of personal computers are that the software structure is simple, the communication load is kept low, and the general operation of the system is kept reasonably flexible.

Ada has been chosen as the language around which the implementation is to be built. It was decided early on that the problems of developing a distributed system would require a language that provided a good degree of abstraction and type control, and since other projects in the department are closely concerned with the use and implementation of the Ada language [3], it seemed the obvious choice. An important result of using Ada is that all interprogram communication is achieved through task rendezvous: kernel objects called 'Mediums' act as buffer tasks, which accept and forward messages.

Achievements

The whole system has been named PULSE. Each machine in the network must run at least a kernel and a file server. The kernel supports the requirements of the Ada language, as well as allowing several programs to run concurrently, and to intercommunicate. It also provides basic management facilities for the allocation and manipulation of new program images. Certain Ada exceptions may be raised remotely by and in programs, and the kernel is responsible for their proper forwarding and responses.

The hierarchical filesystem is implemented by an instance of a file server program running on each PULSE machine. This program is written in Ada, and makes full use of the language's tasking facilities. Each server is responsible for all access to, and management of, files on its machine. This includes the loading of programs, and the association of particular IPC channels with file names.

In order to improve the reliability and availability of file manipulation, a scheme of file replication has been adopted. A particular file may exist at least as a 'master': however, additional, read-only copies known as 'duplicates' may be created by users. The choice of which copy of a file depends, for instance, on whether updating is necessary, whether the master is younger than its duplicates, or whether a particular machine, and hence its disk, is available or not. A full description is given by Tomlinson et al [4].

A basic set of software tools has been written in Ada, in order to test the current system. These include a simple shell which allows programs to run in both foreground and background.

Work in Hand

Work is proceeding at the moment on the automatic updating of duplicate files once their master has been changed. In particular, effort is directed to the ability of a user to set or inhibit the amount of searching performed by the file system. The search map consists of four areas: they are local duplicates, local masters, remote masters, and remote duplicates. The first three categories may be selected, but the last still requires development.

Some experiments are in progress, investigating distributed user programs. For example, no provision is made at a low level for loading programs remotely, so a suite of programs is being developed which can implement this. The servers in this suite associate IPC capabilities with filenames, which in turn are accessed by clients. Another experimental program is a simple mail server, which requires senders to update a common destination file, and then to inform the recipient by means of a named IPC capability.

The project investigator is Dr I C Wand, and the Research Assistants are D Keeffe, G M Tomlinson, and A J Wellings

References

1. A.J.Wellings, I.C.Wand and G.M.Tomlinson, 'Distributed UNIX Project 1981', YCS.47(1981), Department of Computer Science, University of York (21 December 1981).
2. M.V.Wilkes and D.J.Wheeler, 'The Cambridge Digital Communication Ring', *Local Area Communications Networks Symposium,* Mitre Corp. and National Bureau of Standards, Boston (May 1979).
3. J.A.Murdie, 'Functional Specification of the Release 0 Ada Workshop Compiler', YCS.54, Department of Computer Science, University of York, (October 1982).
4. G.M.Tomlinson, D.Keefe, I.C.Wand and A.J.Wellings, 'The PULSE Distributed Filesystem', In course of preparation, University of York (August 1983). r wand.emr 16897 a

York

DR I C WAND
UNIVERSITY OF YORK

UNIX X25 INTERFACE
Sept 79 - Aug 83

Background

The objective of this project, which is funded through the EMR (q.v.) Contract mechanism, is to develop software to link the Unix systems of the DCS research groups via SERCnet or the British Telecom's PSS Network to facilitate more direct cooperation and communication and a greater sharing of the resources.

Achievements

The York UNIX-X25 communications package has been implemented as follows [1,2]: a FALCON SBC-11/21 or LSI-11/02 front-end processor, providing X25 levels 1, 2 (LAPB) and 3, a Transport Service and Host X29; and a suite of user programs running in Unix. The system supports 16 X25 logical channels which are designated (by the user) for use as incoming or outgoing X29/TS29 calls of FTP/MAIL streams.

York have used the COMSYS X25 package written at UCL, to avoid duplication of effort, and have implemented Yellow Book Transport Service and Host X29/TS29 and a protocol to handle data flow between the front-end and Unix. An additional device driver has been provided for Unix.

The user programs include

- X29/TS29 PAD program;
- spooled NIFTP 80 (P and Q processes);
- JNT Mail service;
- access control and call logging facilities.

There is a full on-line documentation for these programs.

Release 1 was distributed late 1982. Release 2 is scheduled for September 1983. It offers much enhanced user programs and greater reliability of the front-end. Release 2.1, which supports the PERQ running PNX, is undergoing field tests and will be available by October 1983.

Future plans include support for the other versions of Unix, integration of the Bristol JTMP software, implementation of ISO transport layers, a high performance front-end software in C (so it can run in non-DEC hardware).

References

1. K.S.Ruttle and I.C.Wand, 'X25-UNIX: Memo Nine, Design Proposal', Department of Computer Science, University of York, March 1980.
2. K.S.Ruttle, 'UNIX-X25 Communications: Release 2 Product Description', Department of Computer Science, University of York, September 1983.

Part III

1. SUMMARY OF ACTIVE RESEARCH GRANTS

	GRANT HOLDER	AWARD K pounds	DURATION	RAs
1	S Abramsky	46.7	Sept 82-Sept 84	1
2	Prof D Aspinall	193.6	Apr 82-Sept 85	9
3	Dr D Bell, Mr N Willis and Dr J Kerridge	60.9	Apr 82-Sept 85	1
4	Dr K H Bennett	30.1	Dec 82-Dec 83	1
5	R Bornat, Prof G F Coulouris and S Abramsky	55.0	Nov 81-Oct 83	1
6	Mrs H Brown, S E Binns and J D Caul	46.7	Jan 82-Jan 85	1
7	Prof P J Brown, Dr P H Welch	40.1	Sept 82-Aug 85	1
8	Prof G D Cain & R C S Morling	77.8	Feb 83-Sept 85	3
9	Prof G F Coulouris	41.2	Oct 81-Oct 83	2
10	R J Cunningham and Dr J Kramer	52.5	Jan 82-Dec 84	1
11	Dr J Darlington	392.2	Oct 82-Oct 85	2
12	Dr L C W Dixon	27.7	Jan 81-Dec 83	1
13	Prof D J Evans, Dr I Newman and Dr M C Woodward	121.6	June 83-May 86	1
14	Prof D J Evans	33.9	Jan 83-Dec 85	1
15	Dr J A Gordon	27.5	March83-Feb 86	1
16	Prof R D Grimsdale, Dr F Halsall	171.4	Oct 79-March84	3
17	Prof R D Grimsdale, Dr F Halsall	59.5	Jan 82-Dec 84	1
18	Prof R L Grimsdale and Dr D J Woollons	25.0	May 83-Dec 83	2
19	Dr J R Gurd, Dr I Watson	398.1	Oct 81-Sept 85	2
20	P L Higginson	53.9	Jan 83-Dec 84	2
21	P L Higginson	40.6	July 83-June 85	1
22	Prof C A R Hoare, J E Stoy and M K Harper	186.9	Jan 82-Dec 84	2
23	J W Hughes and M S Powell	39.5	Oct 83-Sept 86	2
24	D Hutchison, W D Shepherd	57.1	Oct 81-Sept 84	1
25	Dr R N Ibbett, Dr D A Edwards and Dr C J Theaker	58.9	June 83-Oct 84	1
26	Prof J K Iliffe	20.0	Jan 83-Dec 83	1
27	Dr C C Kirkham and Dr J R Gurd	30.1	Mar 81-Feb 84	1
28	Dr P E Lauer	30.5	July 82-June 84	1
29	Dr P E Lauer	10.7	May 83-Apr 84	1
30	N Martin	83.6	July 83-June 85	2
31	Dr I Mitrani & Dr T Betteridge	47.6	Oct 83-Sept 86	1
32	I Page	27.6	Mar 83-Feb 85	1
33	S L Peyton-Jones	46.3	July 83-June 85	1
34	Dr G D Plotkin, A J R G Milner	75.9	Nov 81-Oct 84	1
35	Prof B Randell	271.3	Aug 81-Sept 84*	5
36	Prof B Randell	31.4	Aug 81-Aug 84	1
37	Dr J D Roberts	32.9	March82-Feb 85	1
38	Dr M R Sleep	105.1	Aug 83-July 86	2
39	Dr M S Sloman & Dr J Kramer	128.3	Jan 83-Dec 85	2
40	Dr D G Smith	8.3	Feb 83-Jan 84	1
41	Prof D A Turner	112.1	Jan 83-Dec 86	2
42	W W Wadge	10.9	Oct 83-Sept 84	1
43	Dr I C Wand	130.6	Nov 81-Nov 84	3
	Total	3551.6		

* denotes a 4 year rolling grant. EO denotes a grant which provides funds only for the purchase of equipment

2. INVESTIGATORS AND TITLES OF ACTIVE GRANTS

1. S Abramsky
   An Applicative Programming Methodology for Concurrent and Distributed Systems.

2. Prof D Aspinall
   The Use of Microprocessors in Information Systems.

3. Dr J Bell, Mr N Willis and Dr J Kerridge
   An Evaluation of Alternative Implementations of Multi-processor Communications Mechanisms.

4. Dr K H Bennett
   A Reliable Network-Transparent Filestore for a Collection of Personal Workstations.

5. R Bornat, Prof G F Coulouris and S Abramsky
   Development of a Distributed Pascal-M System.

6. Mrs H Brown, S E Binns and J D Caul
   Typesetting and Text Processing Server for the Cambridge Ring.

7. Prof P J Brown and Dr P H Welch
   Compiling Servers for the Cambridge Ring.

8. Prof G D Cain and R C S Morling
   Local Data Network for Instrumentation Applications.

9. Prof G F Coulouris
   Distributed System Requirements for Effective Man-Machine Interaction

10. R J Cunningham and Dr J Kramer
    Computer Assisted Methods for Developing Verified Software.

11. Dr J Darlington
    Development of a General Purpose Applicative Language Machine and Applicative Language Environment.

12. Dr L C W Dixon
    The Advantages of a Parallel Computation Facility for Global Optimisation

13. Prof D J Evans, Dr I Newman and Dr M C Woodward
    The Relationship between Multi-processor Performance, Workload Distribution and Algorithm Characteristics.

14. Prof D J Evans
    An Investigation of the Relationship between Algorithm Structure and Parallel Architectures.

15. Dr J A Gordon
Secure High Speed Data Transmission.

16. Prof R L Grimsdale, Dr F Halsall
The Design and Implementation of a Multi-microprocessor System.

17. Prof R L Grimsdale and Dr F Halsall
Decentralised Resource Management in a Tightly-Coupled Multicomputer System.

18. Prof R L Grimsdale and Dr D J Woollons
Advanced Techniques for the Computer Generation of Images.

19. Dr J R Gurd and Dr I Watson
A Multilayered Dataflow Computer System.

20. P L Higginson
A Study of Protocol Requirements for Mulit-Media Messages and Intercommunicating Office Machines.

21. P L Higginson
A Study of Protocol Requirements for Multi-Media Messages and Intercommunicating Office Machines (Additional Support).

22. Prof C A R Hoare, J E Stoy and M K Harper
Distributed Computing Software.

23. Mrs J W Hughes and M S Powell
A Cross Development System for Distributed Computing Systems.

24. D Hutchison and W D Shepherd
Direct Comparison of Ring and Ethernet Type Systems.

25. Dr R N Ibbett, Dr D A Edwards and Dr C J Theaker
A High Performance Optical Fibre Based Local Area Computer Network.

26. Prof J K Iliffe
Design Studies for Active Memory Arrays.

27. Dr C C Kirkham, Dr J R Gurd
Development Software for a Prototype Data Flow Computer.

28. Dr P E Lauer
A Computer Based Environment for the Design and Analysis of Highly Parallel and Distributed Computing Systems.

29. Dr P E Lauer
Realisation of a High Level Formalisation in a Computer Based Environment for Designed Distributed Systems.

30. **N Martin**
Message Services and Directory Development.

31. **Dr I Mitrani and Dr T Betteridge**
Modelling and Performance Evaluation of Distributed Unix Systems.

32. **I Page**
Evaluation of the Disarray Graphics Processor Array.

33. **S L Peyton-Jones**
Computer Architectures for Functional Programming.

34. **Dr G D Plotkin and A J R G Milner**
Semantics of Non-Deterministic and Concurrent Computation.

35. **Prof B Randell**
Computer Architectures Supporting General-purpose Highly Parallel Computation.

36. **Prof B Randell**
Reliability and Integrity of Distributed Computing Systems.

37. **Dr J D Roberts**
Transformation of Programs for Tightly Coupled Distributed Systems.

38. **Dr M R Sleep**
Programming an Extensible Array of Transputers using a Naturally Extended Functional Notation.

39. **Dr M S Sloman and Dr J Kramer**
Specification and On-Line Management for Distributed Real-Time Systems.

40. **Dr D G Smith**
Overload Control in Processor-Based Systems.

41. **Prof D A Turner**
An Applicative Language Machine and Operating System.

42. **W W Wadge**
Demand Driven Dataflow.

43. **Dr I C Wand**
Operating System for a Network of Personal Computers.

3. SUMMARY OF GRANTS COMPLETED IN 82/83

	GRANT HOLDER	AWARD K pounds	DURATION	RAs
1	Dr J M Bacon	4.3	June 81-May 83	0
2	Dr K H Bennett	87.0	Oct 79-Dec 82	2
3	Dr F W Burton and Dr M R Sleep	66.0	Aug 81-July 83	2
4	Dr D W Bustard & Dr J Elder	115.0	Oct 79-July 83	3
5	Prof G D Cain and R C S Morling	42.0	May 81-Nov 82	3
6	Dr R Dowsing	68.0	April80-Sept 83	2
7	Prof D J Evans, Dr I A Newman	28.1	July 79-Dec 82	1
8	Prof D J Evans, Dr I A Newman and Dr M C Woodward	126.6	Dec 79-May 83	2
9	Prof J P Fitch and Dr P J Willis	9.2	July 82-June 83	0
10	Dr P W Grant	1.4	Oct 79-Sept 82	0
11	Dr P Henderson	20.8	Oct 81-Sept 83	1
12	J W Hughes and M S Powell	50.8	Oct 80-Sept 83	2
13	Dr J R W Hunter and Dr C S Mellish	12.9	Oct 82-Sept 83	1
14	Prof P T Kirstein	274.2	Oct 78-Feb 83	4
15	Dr I Mitrani	22.8	Apr 81-Mar 83	1
16	Prof R M Needham	28.4	June 80-Sept 82	1
17	I Page	10.4	Mar 82-Feb 83	1
18	Prof B Randell	226.5	Oct 78-Sept 82	5
19	Prof B Randell	69.2	Oct 80-Sept 83	2
20	J A Sharp	2.6	Apr 81-Mar 83	0
21	W D Shepherd	4.7	Oct 81-Sept 82	0
22	Dr M R Sleep	10.5	Jan 80-Dec 82	0
23	W W Wadge	31.2	Oct 80-Sept 83	1
	Total	1312.6		

4. INVESTIGATORS AND TITLES OF GRANTS COMPLETED IN 82/83

1. Dr J M Bacon
An Investigation of Software Structures to Facilitate Distribution.

2. Dr K H Bennett
A Distributed Filestore.

3. Dr F W Burton and Dr M R Sleep
Distributed Evaluation of Applicative Programs on a Highly Interconnected Network.

4. Dr D W Bustard and Dr J Elder
The Design, Implementation and Application of Languages for Distributed Computing.

5. Prof G D Cain and R C S Morling
Local Area Data Network for Instrumentation Applications.

6. Dr R D Dowsing, (Dr P W Grant)
The Specification and Implementation of Programs on a Multi-microprocessor.

7. Prof D J Evans, Dr I A Newman
An Investigation of the Relationship between Algorithm Structure and Parallel Architectures.

8. Prof D J Evans, Dr I A Newman, Dr M C Woodward
A Research Vehicle for Investigating the Use of Closely Coupled Distributed Systems.

9. Prof J P Fitch and Dr P J Willis
A Feasibility Study of a Tree Structured Arrangement of Processors.

10. Dr P W Grant, (Dr R D Dowsing)
The Specification and Implementation of Programs on a Multi-microprocessor.

11. Dr P Henderson
Purely Functional Operating System.

12. J W Hughes, M S Powell
Multi-processor Software Engineering.

13. Dr J R W Hunter and Dr C S Mellish
A Heterogeneous Interactive Distributed Computing Environment for the Implementation of AI Programs.

14. Prof P T Kirstein
Communication Protocols in the Context of X25 Computer Networks.

15. Dr I Mitrani
Modelling and Performance Evaluation for Distributed Computing Systems.

16. Prof R M Needham
Distributed Computing using Wide Band Communications.

17. I Page
DISARRAY-An Evaluation of an Array Processor for Bitmap Displays.

18. Prof B Randell
Reliability and Integrity of Distributed Computing Systems.

19. Prof B Randell
Programming Decentralised Computers.

20. J A Sharp
A Dataflow Program Design Environment - Its Implementation and Use as a Research Vehicle.

21. W D Shephard
Gateways for the Interconnection of Cambridge Rings and Ethernet-like Networks.

22. Dr M R Sleep
The Potential of Pure Combinatory Code for Distributed Computing.

23. W W Wadge
LUCID as a Data Flow Language.

5. **SUMMARY OF PREVIOUSLY COMPLETED RESEARCH GRANTS**

	GRANT HOLDER	AWARD K pounds	DURATION	RAs
1	Prof D Aspinall, Dr R D Dowsing and Dr E L Dagless	90.0	Oct 77-Sept 79	6
2	Dr K H Bennett	6.5	Oct 78-Sept 79	1
3	S E Binns, Prof P J Brown, and Dr E B Spratt	10.1	July 79-June 81	EO
4	R Bornat	22.0	July 80-July 82	1
5	D Coleman, J W Hughes	17.0	Oct 78-Sept 80	1
6	Prof G F Coulouris	81.8	Oct 77-Sept 80	2
7	R J Cunningham and J Kramer	0.6	Dec 79-Nov 80	0
8	Dr J R Gurd, Dr I Watson	79.6	July 78-Sept 81	2
9	Dr F K Hanna	30.5	Oct 78-Sept 81	1
10	S E Hersom	10.0	Oct 79-Sept 81	0
11	Prof C A R Hoare, J E Stoy	81.0	April78-March82	2
12	Prof C A R Hoare	48.6	July 79-March80	EO
13	Dr J R W Hunter Dr K D Baker and Dr A Sloman	72.7	April79-March82	2
14	D Hutchison, W D Shepherd	6.1	Oct 80-Sept 81	1
15	Prof P T Kirstein	10.0	July 79-March80	EO
16	Dr P E Lauer	53.2	Jan 79-June 82	2
17	R M McKeag	5.6	June 80-Oct 81	0
18	A J R G Milner	10.5	Jan 78-Sept 79	1
19	Dr I Mitrani	18.0	Apr 79-Mar 81	1
20	Prof R M Needham	34.0	Aug 78-July 81	1
21	Dr P E Osmon	23.0	July 78-Sept 81	1
22	Dr P E Osmon	6.8	July 79-March80	EO
23	I Page	29.8	July 78-June 80	1
24	I Page	5.2	Sept 81-Feb 82	0
25	Dr Y Paker	31.9	April79-March81	1
26	Dr G D Plotkin, A J R G Milner	39.0	Oct 78-Sept 81	2
27	Prof I C Pyle	60.5	April75-Dec 78	1
28	Prof B Randell	16.6	July 78-Sept 80	1
29	N H Shelness	101.1	Oct 77-Sept 80	3
30	Dr M R Sleep	1.5	Oct 78-Sept 79	0
31	Dr M S Sloman	42.0	Sept 78-Aug 81	1
32	Prof D A Turner	3.0	Mar 80-Mar 82	0
33	W W Wadge	2.3	April79-April80	0
34	Dr I C Wand	20.2	Jan 78-Dec 80	1
35	Dr I C Wand	7.6	July 79-March80	EO
36	Dr I C Wand	41.9	Oct 79-Sept 81	2
37	Dr C Whitby-Strevens, M D May	123.0	Oct 77-April80	3
	Total	1243.2		

EO denotes a grant which provides funds only for the purchase of equipment

6. INVESTIGATORS AND TITLES OF PREVIOUSLY COMPLETED GRANTS

1. Prof D Aspinall, Dr R D Dowsing and Dr E Dagless
   The Use of Microprocessors in Information Processing Systems.

2. Dr K H Bennett
   A Feasibility Study of Loosely Linked Computers.

3. S E Binns, Prof P J Brown, Dr E B Spratt
   Compiling Servers for the Cambridge Ring (at Kent).

4. R Bornat
   Pascal-M: A Language for the Design of Loosely-Coupled Computer Systems.

5. D Coleman, J W Hughes
   Developing a Program Methodology for Multiprograms.

6. Prof G F Coulouris
   Distributed System Requirements for Effective Man-Machine Interaction.

7. R J Cunningham and Dr J Kramer
   Computer Assisted Methods for the Design and Production of Verified Software.

8. Dr J R Gurd, Dr I Watson
   A Ring-Structured Data Flow Computer System.

9. Dr F K Hanna
   Distributed Processing Systems for Interactive Knowledge Bases.

10. S E Hersom
    Development of Optimisation Algorithms for Parallel Computation.

11. Prof C A R Hoare, J E Stoy
    Software Engineering.

12. Prof C A R Hoare
    Workstations for Software Engineering.

13. Dr J R W Hunter, Dr K D Baker, Dr A Sloman
    Interactive Software Tools for Distributed Computing Systems with an Application to Picture Interpretation.

14. D Hutchison, W D Shepherd
    A Feasibility Study into the Complexity of Gateways for Connecting Ethernet to Ring Networks.

15. Prof P T Kirstein
    Communication Protocols in the Context of X25 Computer Networks - Supplementary Proposal.

16. Dr P E Lauer
Design and Analysis of Highly Parallel Distributed Systems.

17. R M McKeag
Collaborative Research into Parallelism

18. A J R G Milner
Applications of Flow Algebras to Problems in Concurrent Computation.

19. Dr I Mitrani
Modelling and Performance Evaluation for Distributed Computing Systems.

20. Prof R M Needham
Development of the Cambridge Ring.

21. Dr P E Osmon
Implementation of a High Level Data Flow Programming Language.

22. Dr P E Osmon
Provision of Facilities for Implementation of a Data Flow Programming Language.

23. I Page
A High Quality Display for Effective Man-Machine Interaction.

24. I Page
DISARRAY-An Array Processor for Bitmap Displays.

25. Dr Y Paker
Computer Aided Multi-microprocessor Systems Modelling, Simulation and Performance Evaluation.

26. Dr G D Plotkin, A J R G Milner
Semantics of Non-Deterministic and Concurrent Computation.

27. Prof I C Pyle
Real Time Programming Languages for Industrial and Scientific Process Control.

28. Prof B Randell
A Project to Investigate the Design of Highly Concurrent General-Purpose Computing Systems.

29. N H Shelness
An Architecture for a Multiple Computer System.

30. Dr M R Sleep
Instruction Sets for Data Flow Architectures: A Comparative Study.

31. Dr M S Sloman
Communications for Distributed Process Control.

32. Prof D A Turner
A denotational language for Dataflow Machines.

33. W W Wadge
Distributed Implementation of Nonprocedural Languages.

34. Dr I C Wand
MODULA Distribution and Promulgation.

35. Dr I C Wand
MODULA Distribution and Promulgation - Supplementary Proposal.

36. Dr I C Wand
Distributed Operating System for Time-Sharing.

37. Dr C Whitby-Strevens, M D May
A Building-Block System for Distributed Computing.

7. SUMMARY OF CO-OPERATIVE RESEARCH GRANTS

	GRANT HOLDER	AWARD K pounds	DURATION	RA's
1	Mrs J W Hughes and M S Powell (with British Leyland)	14.0	Feb 82-Jan 84	1
2	D Hutchison and W D Shephard (with British Telecom)	11.0	July 82-Dec 83	1
3	Prof R L Grimsdale and Dr F Halsall (with Rediffusion)	107.4	Oct 82-Sept 85	1
	Total	132.4		

8. INVESTIGATORS OF CO-OPERATIVE GRANTS AND TITLES

1. Mrs J W Hughes and M S Powell
   (with British Leyland)
   A Distributed Line Monitoring System.

2. D Hutchison and W D Shephard
   (with British Telecom)
   Low-cost X25 Interfaces for Ring and Ethernet-Type Local Networks.

3. Prof R L Grimsdale and Dr F Halsall
   (with Rediffusion)
   The Design and Implementation of a Broad Band Local Area Network.

9. VISITING FELLOWSHIPS

The DCS Programme has obtained the following Visiting Fellowships.

INCOMING

Host	Visiting Fellow	Award K pounds	Duration
Dr D J Cooke Loughborough	Prof K Lautenbach Bonn	0.2	April80-April80
Dr D J Cooke Loughborough	Prof E Gelenbe Paris	0.2	April80-April80
Dr D J Cooke Loughborough	Dr W Kwasowiec Warsaw	0.4	April80-April80
Prof J P Fitch Bath University	Dr J B Marti Utah	7.0	June 82-Dec 82
Prof C A R Hoare Oxford University	J R Abrial Paris	7.2	Sept 79-Mar 80
Prof C A R Hoare Oxford University	J R Abrial Paris	4.0	Mar 80-May 80
Prof C A R Hoare Oxford University	Dr C C Morgan Sydney University	4.0	Jan 82-Mar 82
Dr P E Lauer	Prof R H Campbell	3.2	May 81-Aug 81
Dr P E Lauer Newcastle University	Prof R Janicki Warsaw	6.3	Mar 82-Aug 82
Dr I Mitrani Newcastle University	Prof E Gelenbe Paris	5.0	June 80-Oct 80
Prof R M Needham Cambridge University	Dr D D Clarke MIT Laboratory	2.0	Feb 82-May 82
Prof R M Needham Cambridge University	Dr J G Mitchell Xerox PARC	4.1	Oct 80-Mar 81
Prof Y Paker Central London Poly	Prof E Jensen Honeywell	0.8	Aug 82-Sept 82
Dr G D Plotkin Edinburgh University	Dr P D Mosses Denmark	19.4	Jan 83-Dec 83
Prof B Randell Newcastle University	Prof R H Campbell	9.2	Sept 82-Aug 83
Dr J D Roberts Reading University	Prof K Lautenbach Bonn	0.4	Mar 82-April82

Dr J D Roberts Reading University	Prof S Owicki Stanford University	0.2	July 79-July 79
Mr W W Wadge Warwick University	Prof E A Ashcroft Waterloo University	1.0	Feb 82-Mar 82
Dr C Whitby-Strevens Warwick University	Prof E Balkovich Connecticut	2.7	June 79-Aug 79
Dr C Whitby-Strevens Warwick University	E D Jensen Honeywell	0.1	Oct 78-Oct 78

OUTGOING
-none

10. EMR CONTRACTS

10.1. Introduction

The aims of the DCS Programme include the reduction of duplicated effort and the production of software and hardware which are of general benefit to the majority of DCS investigators. Often the production of such items is more of a development task than a pure research exercise. In such cases, the DCS Panel can ask the Rutherford and Appleton Laboratories to draw up a suitable contract between the Laboratories and a university or industrial company to develop such a specific product. These agreements are called EMR (Extra Mural Research) contracts. The projects of Dr Bustard and Dr Wand are developing software products. These projects are described under the entries for Queen's University of Belfast and the University of York respectively. The EMR to Dr Gurd for a User Liaison Officer provides an additional staff post with the Manchester Dataflow Project. The terms of reference of the User Liaison Officer are to foster awareness and usage of the Dataflow Machine. The User Liaison Officer is responsible to the DCS Panel.

10.2. EMR's Awarded

EMR HOLDER	PROJECT	AWARD K pounds	DURATION
Dr D Bustard	Pascalplus compiler	10.0	Nov 78-Sept 80
Dr D Bustard	Pascalplus compiler	23.0	Apr 81-Sept 84
Software Sciences Ltd	Pascalplus tools	16.0	Dec 78-Feb 79
Dr I Wand	Unix X25	25.0	Sept 79-Nov 81
Dr I Wand	Unix X25	13.8	Nov 81-Dec 83
Dr J R Gurd	User Liaison Officer	27.7	Jan 83-Dec 84

11. MEETINGS

11.1. Introduction

The DCS Panel sponsors a series of meetings to promote cooperation between individual DCS funded researchers and communication between DCS funded projects and industry, Government Establishments and other research groups.

11.2. Meetings to Date

Date	Subject	Host	Type
May 78	Real Time	York	SIG
May 78	Ironman	York	Workshop
June 78	Dataflow	Newcastle	Workshop
Aug 78	Dataflow	RL	SIG
Sept 78	Languages	Warwick	Workshop
Oct 78	'Jensen'	Warwick	SIG
Jan 79	'Dijkstra'	Oxford	Workshop
Jan 79	Academic DCS	IEE	Colloquium
Apr 79	Networks	UCL	SIG
May 79	Industrial DCS	IEE	Colloquium
July 79	Ada	York	Workshop
July 79	'Bowles'	Oxford	Workshop
Oct 79	Applicative Langs	Newcastle	Workshop
Apr 80	BCS FACS	Loughborough	Workshop
Sept 80	Cambridge Ring	UCL	SIG
Oct 80	Applicative Langs	Newcastle	Workshop
Feb 81	Ring Protocols	UCL	SIG
Mar 81	Ring/Pascal	Keele	SIG
Mar 81	Dist.Filestore	Keele	SIG
Mar 81	Ring Simulation and Modelling	Cambridge	SIG
Apr 81	Distributed Oper Systems	QMC	SIG
Apr 81	VLSI	Newcastle	Workshop
Apr 81	Ring Protocols	Cambridge	SIG
May 81	'Liskov'	Oxford	Workshop
June 81	Closely Coupled Systems	Sussex	Workshop
July 81	Dataflow	I.C.	SIG
July 81	'Herzog'	Loughborough	Workshop
July 81	Neptune Seminar	Loughborough	Workshop
Sept 81	Closely Coupled Systems	UMIST	SIG
Sept 81	CYBA-M (Ind Presentation)	UMIST	Workshop
Feb 82	Logic Programming	I.C.	Workshop
Apr 82	BCS FACS	Reading	Workshop
Apr 82	Closely Coupled Systems	Loughborough	SIG

Date	Subject	Host	Type
June 82	Cambridge Ring	QMC	SIG
Nov 82	Formal Methods	Oxford	Workshop
Dec 82	Closely Coupled Systems	Bristol	SIG
Dec 82	Distributed Operating Systems	York	SIG
Apr 83	Declarative Programming	UCL	Workshop
Apr 83	Specification (FACS)	York	Workshop
June 83	High-Speed Computer Architectures	Manchester	SIG

The Loughborough Reading and York FACS meetings were organised by the British Computer Society Formal Aspects of Computer Science Special Interest Group in association with the DCS Programme.

11.3. Conferences

The following conferences have been organised. Copies of the proceedings of the Industrial Conference are available from the Industrial Coordinator.

Date	Subject	Host
July 82	DCS Conference	Strathclyde
Mar 83	Industrial Conference	NCC, Manchester
June 83	DCS Conference	Sussex

12. DCS EQUIPMENT POOL

Equipment in the pool is loaned to investigators following authorisation from the Panel. The pool serves four functions:

1. To provide communications facilities between the research groups in the Programme. Initially magnetic tape was the recognised communications medium, but PSS and SERCnet now play a key role.
2. To provide communications facilities within research groups. The pool contains Cambridge Ring equipment for this purpose.
3. To provide commercial UNIX licences and other commonly used software.
4. To provide a bank of LSI-11, Teraks and more recently PERQ computer systems. By sharing these processors several projects can benefit from a large resource which could not be provided to each of them.

The equipment pool is a good example of the benefits of a coordinated research programme in the realm of cost-effective usage of equipment.

Equipment Details

Item	Quantity
Newbury VDU	10
EMI tape decks	10
Diablo printers	11
Dicoll LSI-11s	15
RT-11 kits	5
UNIX licences	14
Modems (rented)	5
'Teraks'	10
UCSD Pascal source code kits	15
Cambridge Ring equipment	16
X25 front-ends	8
ICL Perqs	6

13. ADDRESSES

13.1. Coordination Team

Dr D A Duce
Academic Coordinator, DCSP
Computing Division
Rutherford Laboratory
Chilton
Didcot
Oxon
OX11 0QX
0235-44551 or 21900 Ext 5511

Mr F B Chambers
Industrial Coordinator, DCSP
Logica Ltd
64 Newman Street
London
W1A 4SE
01-636-5440 Ext 242

Miss G P Jones
Technical Secretary, DCSP
Computing Division
Rutherford Laboratory
Chilton
Didcot
Oxon
OX11 0QX
0235-446147 or 0235-21900 Ext 6147

M Hotchkiss
Scientific Secretary
Distributed Computing Systems Panel
Science and Engineering Research Council
P O Box 18
Swindon
Wilts
SN1 5BW
0793-26222 Ext 2260

13.2. Grant Holders

S Abramsky
Department of Computing
Imperial College
180 Queens Gate
South Kensington
London
SW7 2BZ
01-589-55111

Prof D Aspinall
Dept of Computation
University of Manchester
Institute of Science and Technology
P O Box 88
Manchester
M60 1QD
061-236-3311

Dr J M Bacon
School of Information Sciences
The Hatfield Polytechnic
P O Box 109
College Lane
Hatfield
Herts
AL10 9AB
070-72-68100

Dr K D Baker
(late University of Sussex)
Dept of Electronic & Electrical Engineering
Plymouth Polytechnic
Drakes Circus
Plymouth
PL4 8AA
0752-264568

Dr D H Bell
Sheffield City Polytechnic
Computer Studies Department
Pond Street
Sheffield
S1 1WB
0742-20911

Dr K H Bennett
Dept of Computer Science
University of Keele
Keele
Staffs
ST5 5BG
0782-621111 Ext 583

Dr T Betteridge
Computing Laboratory
University of Newcastle upon Tyne
Claremont Tower
Claremont Road
Newcastle upon Tyne
NE1 7RU
0623-329233

S E Binns
Computing Laboratory
The University
Canterbury
Kent
CT2 7NF
0227-66822 Ext 7616

R Bornat
Dept of Computer Science and Statistics
Queen Mary College
Mile End Road
London
E1 4NS
01-980-4811

Mrs H Brown
Computing Laboratory
The University
Canterbury
Kent
CT2 7NF
0227-66822 Ext 7550

Prof P J Brown
Computing Laboratory
The University
Canterbury
Kent
CT2 7NF
0227-66822 Ext 636

Dr F W Burton
School of Computing Studies
University of East Anglia
University Village
Norwich
NO4 8BC
0603-56161

D W Bustard
Department of Computer Science
The Queen's University of Belfast
Belfast
BT7 1NN
0232-45133

Prof G D Cain
Engineering Division
Polytechnic of Central London
115 New Cavendish Street
London
W1M 8JS
01-636-2074

D J Caul
Computing Laboratory
The University
Canterbury
Kent
CT2 7NF
0227-66822 Ext 635

Dr K Clarke
Department of Computing and Control
Imperial College
180 Queens Gate
London
SW 7
01-589-5111

D Coleman
Dept of Computation
University of Manchester
Institute of Science and Technology
P O Box 88
Manchester
M60 1QD
061-236-3311 Ext 2341

Prof G F Coulouris
Dept of Computer Science and Statistics
Queen Mary College
Mile End Road
London
E1 4NS
01-980-4811 Ext 751

R J Cunningham
Department of Computing and Control
Imperial College
180 Queen's Gate
London
SW 7
01-589-5111 Ext 2745

Prof E Dagless
Department of Micro Electronics
Queens Building
University of Bristol
University Walk
Bristol
BS8 1TH
0272-24161

Dr J Darlington
Department of Computing
Imperial College
180 Queens Gate
London
SW 7
01-598-5111

Dr L C W Dixon
School of Information Sciences
P O Box 109
College Lane
Hatfield
Herts
AL10 9AB
070-72-68100 Ext 341

Dr R D Dowsing
School of Computing Studies
The University of East Anglia
Norwich
NR4 7TJ
0603-56161 Ext 204

Dr D A Edwards
Dept of Computer Science
University of Manchester
Oxford Road
Manchester
M13 9PL
061-273-7121

Dr J Elder
Department of Computer Science
The Queen's University of Belfast
Belfast
BT7 1NN
0232-45133

Prof D J Evans
Dept of Computer Studies
University of Technology
Loughborough
Leicestershire
LE11 3TU
0509-63171 Ext 297

Prof J P Fitch
School of Mathematics
University of Bath
Claverton Down
Bath
BA2 7AY
0225-61244

Dr J A Gordon
Dept of Electrical Engineering
The Hatfield Polytechnic
Hatfield
Herts
AL10 9AB
070-72-68100

Dr P W Grant
Dept of Computer Science
University College of Swansea
Singleton Park
Swansea
SA2 8PP
0793-205678

Prof R L Grimsdale
School of Eng. and Applied Sciences
University of Sussex
Falmer
Brighton
BNl 9QT
0273-606755 Ext 74

Dr J R Gurd
Dept of Computer Science
University of Manchester
Oxford Rd
Manchester
M13 9PL
061-273-1993

Dr F Halsall
School of Eng. and Applied Sciences
University of Sussex
Falmer
Brighton
BN1 9QT
0273-606755

Dr F K Hanna
The Electronics Laboratory
University of Kent at Canterbury
Canterbury
Kent
CT2 7NF
0227-66822 Ext 396

M K Harper
Oxford University Computing Laboratory
Programming Research Group
8 Keble Road
Oxford
0865-54141

Prof P Henderson
(late University of Oxford)
Dept of Computer Science
University of Sterling
Sterling
FK9 4LA
0786 3171

P L Higginson
Dept of Computer Science
University College
Gower Street
London
WC1E 6BT
01-387-7050

Prof C A R Hoare
Oxford University Computing Laboratory
Programming Research Group
8 Keble Rd
Oxford
0865-54141

Mrs J W Hughes
Dept of Computation
UMIST
Sackville Street
Manchester
M60 1QD
061-236-3311

Dr J R W Hunter
School of Eng and Applied Sciences
University of Sussex
Falmer
Brighton
BN1 9QT
0273-606755 Ext 133

D Hutchison
(late University of Strathclyde)
Dept of Computing Studies
University of Lancaster
Bailrigg
Lancaster
LA1 4YN
0524-65201

Dr R N Ibbett
Dept of Computer Science
University of Manchester
Oxford Road
Manchester
M13 9PL
061-273-7121

Prof J K Illiffe
Dept of Computer Science
Queen Mary College
Mile End Road
London
E1 4NS
01-980-4811

Dr J M Kerridge
Department of Computer Studies
Sheffield City Polytechnic
Pond Street
Sheffield
S1 1WB
0742-20911

Dr C C Kirkham
Dept of Computer Science
University of Manchester
Oxford Rd
Manchester
M13 9PL
061-273-1993

Prof P T Kirstein
Dept of Computer Science
University College
Gower St
London
WC1E 6BT
01-387-7050 Ext 815

Dr J Kramer
Department of Computing and Control
Imperial College
180 Queen's Gate
London
SW7
01-589-5111

Dr P E Lauer
Computing Laboratory
University of Newcastle upon Tyne
Claremont Tower
Claremont Rd
Newcastle upon Tyne
NE1 7RU
0632-329233

N Martin
Dept of Computer Science
University College
Gower Street
London
WC1E 6BT
01-387-7050

M D May
(late University of Warwick)
INMOS Ltd
Whitefriars
Lewins Mead
Bristol
0272-290861

R M McKeag
Department of Computer Science
The Queen's University
Belfast
BT7 1NN
0232-45133 Ext 3237

Dr C S Mellish
School of Engineering & Applied Science
University of Sussex
Falmer
Brighton
Sussex
BN1 9QT
0273-606755

A J R G Milner
Dept of Computer Science
James Clerk Maxwell Building
The King's Buildings
Mayfield Rd
Edinburgh
EH9 3JZ
031-667-1081 Ext 2753

Dr I Mitrani
Computing Laboratory
University of Newcastle upon Tyne
Claremont Tower
Claremont Rd
Newcastle upon Tyne
NE1 7RU
0632-329233 Ext 271

R C S Morling
School of Engineering & Science
Polytechnic of Central London
115 New Cavendish Street
London
W1M 8JS
01-486-5811

Prof R M Needham
Computer Laboratory
University of Cambridge
Corn Exchange St
Cambridge
CB2 3QG
0223-352435

Dr I Newman
Dept of Computer Studies
University of Technology
Loughborough
Leicestershire
LE11 3TU
0509-63171

Dr P E Osmon
Westfield College
University of London
Kidderpore Av
Hampstead
London
NW3 7ST
01-435-7141 Ext 521

I Page
Dept of Computer Science and Statistics
Queen Mary College
Mile End Road
London
E1 4NS
01-980-4811 Ext 751

Prof Y Paker
School of Engineering & Science
Polytechnic of Central London
115 New Cavendish Street
London
W1
01-486-5811 Ext 6271

S L Peyton-Jones
Dept of Computer Science
University College
Gower Street
London
WC1E 6BT
01-387-7050

Dr G D Plotkin
Dept of Computer Science
James Clark Maxwell Building
The King's Buildings
Mayfield Road
Edinburgh
EH8 3JZ
031-667-1081 Ext 2775

M S Powell
Dept of Computation
UMIST
Sackville Street
Manchester
M60 1QD
061-236-3311

Prof I C Pyle
Dept of Computer Science
University of York
Heslington
York
YO1 5DD
0904-59861 Ext 485

Prof B Randell
Computing Laboratory
University of Newcastle upon Tyne
Claremont Tower
Claremont Rd
Newcastle upon Tyne
NE1 7RU
0632-329233 Ext 235

Dr J D Roberts
The University of Reading
Dept of Computer Science
Whiteknights Park
Reading
RG6 2AX
0734-85123

J A Sharp
Dept of Computer Science
University College of Swansea
Singleton Park
Swansea
SA2 8PP
0792-205678

N H Shelness
(late University of Edinburgh)
British National Oil Corp
150 St Vincent Street
Glasgow
G2 5W
041-204-2525

Prof W D Shepherd
(late University of Strathclyde)
Dept of Computing Studies
University of Lancaster
Bailrigg
Lancaster
LA1 4YN
0524-65201

Dr M R Sleep
School of Computing Studies
University of East Anglia
University Village
Norwich
N04 8BC
0603-56161 Ext 205

Dr A Sloman
School of Social Sciences
University of Sussex
Falmer
Brighton
BN1 9QN
0273-606755 Ext 1012 or 1002

Dr M Sloman
Dept of Computing and Control
Imperial College
London
SW7 2BZ
01-589-5111 Ext 2732 or 2740

Dr D G Smith
Dept of Electronic Science
University of Strathclyde
Richmond Street
Glasgow
G1 1XH
041-552-4400

Dr E B Spratt
Computer Laboratory
The University
Canterbury
Kent
CT2 7NF
0227-66822

J E Stoy
Oxford University Computing Laboratory
Programming Research Group
8 Keble Road
Oxford
0865-54141

Dr C J Theaker
Dept of Computer Science
University of Manchester
Oxford Road
Manchester
M13 9PL
061-273-7121

Dr P C Treleaven
Computing Laboratory
University of Newcastle upon Tyne
Claremont Tower
Claremont Rd
Newcastle upon Tyne
NE1 7RU
0632-329233

Prof D A Turner
Computing Laboratory
University of Kent
Canterbury
CT2 7NF
0227-66822 Ext 7565

W W Wadge
Computer Science Department
University of Warwick
Coventry
CV4 7AD
0203-24011 Ext 2647

Dr I C Wand
Dept of Computer Science
University of York
Heslington
York
YO1 5DD
0904-59861 Ext 5570

Dr I Watson
Dept of Computer Science
University of Manchester
Oxford Road
Manchester
M13 9PL
061-273-1993

Dr P H Welch
Computing Laboratory
The University
Canterbury
Kent
CT2 7NF
0227-66822

Prof J Welsh
Dept of Computation
University of Manchester
Institute of Science and Technology
P O Box 88
Manchester
M60 1QD
061-236-3311

Dr C Whitby-Strevens
(late University of Warwick)
INMOS Ltd
Whitefriars
Lewins Mead
Bristol
0272-290861

N Willis
Computer Studies Department
Sheffield City Polytechnic
Pond Street
Sheffield
S1 1WB
0742-20911

Dr P J Willis
School of Mathematics
University of Bath
Claverton Down
Bath
BA2 7AY
0225-61244

Dr D J Woollons
School of Engineering & Applied Science
The University of Sussex
Falmer
Brighton
Sussex
BN1 9QT
0273-606755

Dr M C Woodward
Dept of Computer Studies
University of Technology
Loughborough
Leicestershire
LE11 3TU
0509-63171

14. DISTRIBUTED COMPUTING SYSTEMS PANEL MEMBERSHIP

PANEL MEMBERS 1977-1978

I M Barron	Chairman
Prof D Aspinall	
A C Conway	Department of Industry
Dr F K Hanna	
D Hebditch	
Prof C A R Hoare	
C J Hughes	
J McNeil	
Prof I C Pyle	
Prof B Randell	
M G Shortland	Control Engineering Committee
Dr R Barrett	Electrical and Systems Engineering Committee
R Landeryou	Secretary, CSERB

PANEL MEMBERS 1979-80

Prof I C Pyle	Chairman
Prof D Aspinall	
Dr R Barrett	
R Brown	Department of Industry
S L H Clark	
Dr F K Hanna	
Prof C A R Hoare	
C J Hughes	
Prof R M Needham	
D Pearce	
Prof M Rogers	

PANEL MEMBERS 1978-79

Prof I C Pyle	Chairman
Prof D Aspinall	
R Brown	Department of Industry
Dr F K Hanna	Control Engineering Committee
D Hebditch	
Prof C A R Hoare	
C J Hughes	
D Pearce	
Prof M Rogers	
G Scarrott	
Dr R Barrett	Electrical and Systems Engineering Committee

PANEL MEMBERS 1980-81

Prof R M Needham	Chairman
N Bernard	Department of Industry
B Brinkman	
Prof R L Grimsdale	
C J Hughes	
A J R G Milner	
R Newey	
D Pearce	
E C P Portman	
Prof M Rogers	

PANEL MEMBERS 1981-2

R Needham	Chairman
B Brinkman	
K Dixon	
Prof R L Grimsdale	
E B Holloway	
A J R G Milner	
R Newey	
E C P Portman	
Dr I C Wand	
M Wright	Department of Industry

PANEL MEMBERS 1982-3

R Newey	Chairman
B Brinkman	
K Dixon	
Prof R L Grimsdale	
E B Holloway	
A J R G Milner	
E C P Portman	
Dr M R Sleep	
Dr I C Wand	
M Wright	Department of Industry

PANEL MEMBERS 1983-4

R Newey	Chairman
B Brinkman	
K Dixon	
Prof R L Grimsdale	
E B Holloway	
A J R G Milner	
E C P Portman	
Dr M R Sleep	
Dr I C Wand	